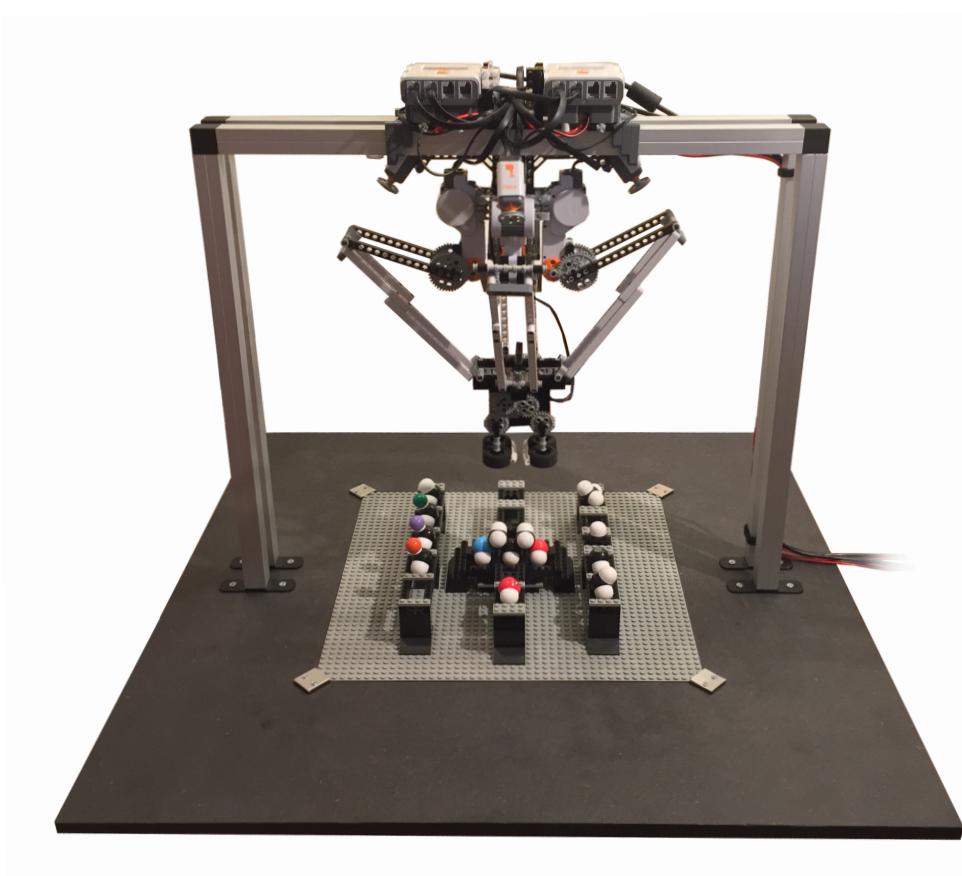


ΔELTAFOUR

Ein Delta-Roboter setzt Molekülmodelle zusammen



Maturitätsarbeit im Themengebiet der Mechatronik

von Victor Klemm, Klasse 6e

Kantonsschule Hohe Promenade, Gymnasium, Zürich

Schuljahr 2014/2015

Name der betreuenden Lehrperson: Michael Liebich

Name der Korreferentin: Dr. Elisabeth Ruh

Inhaltsverzeichnis

1. Vorwort.....	3
2. Einleitung	4
3. Themenwahl	5
4. Konstruktion.....	6
4.1 Konstruktion der Plattform und der Aluminium-Träger	6
4.2 Konstruktion des Delta-Roboters	6
4.3 Molekülsystem.....	11
4.4 Probleme und Lösungen	13
5. Programmierung der Software mit LabVIEW.....	17
5.1 Einleitung	17
5.2 LabVIEW als graphische Programmiersprache	17
5.3 Die wichtigsten Elemente von LabVIEW für das Programm	18
5.4 Übungsprogramme	20
5.5 Erklärung des endgültigen Programms.....	21
5.6 Anpassung des Programms auf die Molekülpositionen.....	27
6. Diskussion	28
7. Zusammenfassung	29
8. Schlusswort	30
9. Dank	30
10. Verwendete Anwendungen	31
11. Literatur- und Quellenverzeichnis, Nachschlagewerke	32
12. Rechtshinweis	36
13. Unterzeichnete Erklärung.....	36
14. Anhang	37
14.1 Ausdruck des gesamten Programms	37
14.2 Tabelle der baubaren Moleküle	37
14.3 Entwürfe und Zeichnungen	38

1. Vorwort

Technische Konstruktionen haben mich schon immer fasziniert. Das LEGO System mit seiner Vielfalt an Bauelementen hat sich bisher in jedem Alter geradezu als ideal angeboten. Das Konstruieren und Modifizieren von Mechanismen, Maschinen und Fahrzeugen macht und macht mir immer noch ausnehmend viel Spass.

Weihnachten 2007 bescherte mir das LEGO Mindstorms NXT-Set. Mit zehn Jahren war ich noch zu jung, um das volle Potential eines programmierbaren Mini-Computers mit Motoren und Sensoren zu erfassen.

Vor zwei Jahren (2012) wurden mein Klassenkollege Dominik Mannhart (der die Faszination für Roboter, Maschinen und vor allem auch für LEGO mit mir teilt) und ich von unserem Mathematiklehrer Dr. Stefan Grieder auf einen Robotik-Wettbewerb von Helveticrobot hingewiesen, bei dem man mit einem innerhalb von 24 Stunden selbstkonstruierten LEGO-Roboter verschiedene Parcours absolvieren musste. Als Team haben wir sowohl im Januar 2013 als auch 2014 in Chur bei diesem spannenden Wettbewerb mitgemacht. Dabei habe ich das vielfältige Potential des NXTs neu entdeckt. Darüber hinaus hat es mir ausserordentlich Spass gemacht, einen Roboter selber zu konzipieren, zu konstruieren und zu programmieren, um eine vorgegebene Aufgabe zu bewältigen.

Die Passion für LEGO und die Erfahrungen in Chur gab den endgültigen Impuls, das Thema für meine Maturitätsarbeit an der Kantonsschule Hohe Promenade Zürich aus dem Themenbereich der Mechatronik zu wählen.



Abb. 1: Robotikwettbewerb in Chur 2014

2. Einleitung

Das Ziel meiner Arbeit ist, einen Delta-Roboter zu konzipieren, zu konstruieren und zu programmieren. Der Roboter soll mittels eines Computers gesteuert werden, welcher mit dem Roboter verbunden ist und mit diesem interagiert. Der Benutzer soll auf dem Computer über ein GUI¹ Moleküle auswählen können, die der Roboter dann automatisch aus verschiedenen, vorgegebenen Molekül-Teilen zu einem Molekülmodell zusammenbaut. Dieses Ziel habe ich am 4. Juli 2014 festgelegt. Anfangs musste ich mich mit folgenden Themen auseinandersetzen:

- Wie baue ich einen stabilen und funktionstüchtigen Delta-Roboter?
- Wie soll der Roboter die Molekül-Teile greifen und wieder absetzen können?
- Wie programmiere ich den Roboter und wie soll das GUI aussehen?
- Welche Art von Molekülmodell verwende ich? Stelle ich es allenfalls selber aus bemalten Holz- oder Styroporkugeln her ?
- Wie baue ich eine Plattform und ein passendes Träger-Gestell, damit der Roboter die Molekül-Teile von oben greifen kann?

Genauere Spezifikationen, wie z.B. wie viele und welche Moleküle der Roboter am Schluss bauen soll, liess ich offen, um sie nach Bedarf auf die Funktionalität des Roboters anpassen zu können.

In dieser Dokumentation wird der Entwicklungsprozess meiner Arbeit beschrieben, die Probleme thematisiert und das Endprodukt erklärt.

¹GUI steht für Graphical User Interface, auf Deutsch: graphische Benutzeroberfläche

3. Themenwahl

In diesem Kapitel werde ich den Prozess meiner Themenwahl beschreiben.

Bereits kurz nach Weihnachten 2013 hatte ich mich mit Herrn Liebich in Verbindung gesetzt, weil er als Robotik-Experte regelmässig ein Robotik-Freifach anbietet. Herr Liebich versicherte mir, dass er mich gerne betreuen würde und somit habe ich mit ihm einen Termin vor den Frühlingsferien 2014 vereinbart.

Durch meine Begeisterung für den Rubik's Cube erwog ich als Erstes einen Roboter zu konstruieren, der eine tetraedrische Version des Rubik's Cubes (Pyraminx) lösen sollte. Für den original 3x3x3 Rubik's Cube existierte nämlich bereits ein Roboter². Wir haben die Idee besprochen und waren beide der Meinung, dass ein solcher Roboter zu wenig originell sei, einigten uns aber darüber, dass meine Maturitätsarbeit mit Robotern zu tun haben sollte.

Kurz darauf stiess ich zufällig auf einen Delta-Roboter und war so fasziniert, dass ich sofort begonnen habe, eine eigene Version mit meinem LEGO Mindstorms NXT zu bauen und zu programmieren.

Nach den Frühlingsferien 2014 habe ich Herrn Liebich einige meiner neuen Roboter-Ideen vorgestellt, z.B. hatte ich die Idee, einen automatischen Tablettenspender zu konstruieren, in welchen das Pflegepersonal die Zeiten und Mengen der Medikamente eintippen könnte. Leider war ein ähnliches Prinzip bereits auf dem Markt³. Auch habe ich ihm einen Film meines funktionstüchtigen Delta-Roboter-Prototypen gezeigt und daraufhin hat er mir geraten, mich unbedingt für meine Maturitätsarbeit weiter mit diesem Robotertyp zu befassen. Sofort war ich für die Weiterentwicklung motiviert. Nun galt es, eine einzigartige Anwendung für den Roboter zu finden. Nach einer eingehenden Recherche mussten wir praktisch alle Anwendungs-Ideen, wie z.B. einen Barkeeper-Roboter oder einen Casino-Dealer-Roboter, wieder verwerfen, da es solche bereits in sehr professioneller Ausführung gab. Es blieb noch eine Idee übrig, deren erfolgreiche Umsetzung zuerst als kaum möglich erschien:

Das automatische Zusammensetzen verschiedener Molekülmodelle, welche mit Hilfe einer Software auf einem Computer vorher ausgewählt werden können.

Bis anhin waren mir nur Molekülmodelle mit einem mühsamen Stäbchen-Stecksystem bekannt, dessen Zusammenbauen durch einen selbstgebauten Roboter nicht möglich war. Trotzdem liess mich die Idee nicht los. Glücklicherweise schlug Herr Liebich das sogenannte Kalottenmodell vor. Nach einigen Überlegungen kam ich auf die Idee, dass die einzelnen Atom-Kugeln mit darin versenkten Magneten zusammenhalten könnten. Somit müsste der Roboter "nur" mit kugelförmigen Objekten. Die Umsetzung der Idee war vielleicht doch möglich. Nach einigen Skizzen und einem weiteren Gespräch stand der Entschluss fest, die Herausforderung anzunehmen.



Abb. 2: Pyraminx

²CubeStormer II

https://www.youtube.com/watch?v=_d0LfkIut2M

4. Konstruktion

Die Konstruktion des Modells lässt sich in drei Einheiten gliedern: Plattform und Aluminium-Träger, Delta-Roboter und Molekülsystem.

4.1 Konstruktion der Plattform und der Aluminium-Träger

Konstruktion der Plattform

Die Anforderungen an die Basis-Plattform waren die folgenden: Sie durfte nicht zu gross und nicht zu schwer sein, sie musste eine gewisse Dicke haben, um Schrauben zu befestigen und ästhetisch zum Roboter passen. Eine schwarz durchgefärbte mitteldichte Holzfaserplatte (MDF-Platte) mit den Massen 75x75x1,5 cm erfüllte diese Anforderungen.

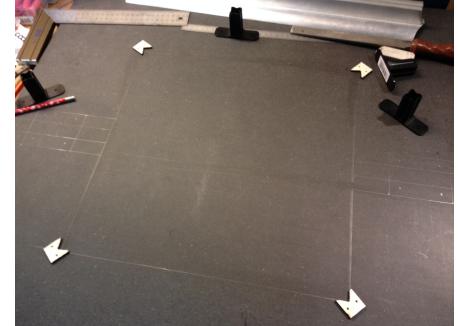


Abb. 3: MDF-Platte

Aluminium-Träger

Die Idee der Aluminium-Träger ergab sich zufällig in den Sommerferien in einem Italienischen Baumarkt, in der Form des Alfer Combitech Systems: ein Steck-System aus zuschneidbaren Aluminium-Stangen mit quadratischem Querschnitt und Verbindungs- und Verankerungselementen aus Kunststoff (Ecken und Füßchen).



Abb. 4: Alfer Combitech System

Für die Konstruktion wurden als Erstes die Stangen auf die richtige Länge zugesägt (vier vertikale Stangen zu 48,5 cm und zwei horizontale zu 54,5 cm). Mit vier Eckteilen wurden daraus zwei parallele Rahmen im Abstand von 2,4 cm zusammengesetzt und diese wiederum mit vier Füßchen an der Plattform festgeschraubt.

4.2 Konstruktion des Delta-Roboters

Was ist ein Delta-Roboter?

Ein Delta-Roboter ist ein Roboter, welcher normalerweise aus drei beweglichen Armen besteht, die an ihren unteren Endpunkten aneinander befestigt sind. Die Arm-Konstruktion entspricht verschiebbaren Parallelogrammen, weshalb ein Delta-Roboter auch als ein Parallel-Roboter bezeichnet wird. Die Bewegung jedes Armes wird aufgrund einer Winkelveränderung durch einen jeweiligen Motor individuell gesteuert. Delta-Roboter werden in der Industrie aufgrund ihrer Geschwindigkeit und Präzision sehr geschätzt.

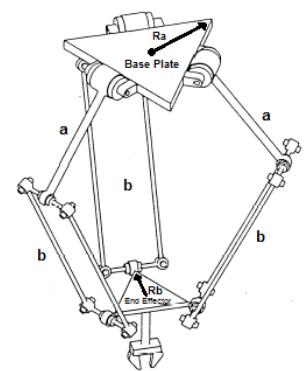


Abb. 5: Bauplan eines gewöhnlichen Delta-Roboters

LEGO-Hardware

Um den Roboter zu konstruieren, wurde der Mini-Computer Mindstorms NXT von LEGO verwendet. Dieses Set besteht aus drei Servomotoren und vielen verschiedenen Sensoren. Da alle Motoren, Sensoren und der NXT-Block mit dem LEGO Technik System kompatibel sind, war es naheliegend, auch den Rest des Roboters aus LEGO zu bauen. Das LEGO System hat auch den Vorteil, dass man den Roboter jederzeit umbauen kann, was sich während des Konstruktions-Prozesses als außerordentlich nützlich erwies, da die mechanische Funktionsweise des Roboters, wie z.B. das Greifsystem oder die beweglichen Arme, immer wieder verändert werden mussten.

Nachfolgend werden die wichtigsten und speziellsten LEGO-Bausteine erläutert und ihre Funktion für den Roboter erklärt.

Der LEGO Mindstorms NXT

Der Mindstorms NXT ist ein programmierbarer Mini-Computer von LEGO. Um den Roboter zu steuern, interagieren zwei Mindstorms NXTs mit einer graphischen Entwicklerplattform namens LabVIEW auf dem Computer. Sie senden Sensordaten und erhalten Befehle, z.B. wann welche Motoren aktiviert werden sollen. Somit bilden sie das Gehirn des Roboters. Jeder NXT-Block verfügt über:

- drei Ausgangssports, welche die Motoren mit Strom versorgen,
- vier Eingangssports, die mit vielen verschiedenen Sensoren verbunden werden können,
- vier Funktions-Tasten,
- ein Display,
- einen USB-Anschluss.

Ein NXT Block kann mit einem Computer entweder über Bluetooth (was mit meinem privaten MacBook Pro leider nicht funktioniert), oder über ein USB-Kabel verbunden werden. Da fünf unabhängige Motoren für das Roboter-Modell benötigt wurden, vier für die beweglichen Arme und einen für das Greifsystem, mussten zwei NXT-Blöcke verwendet werden.



Abb. 6: Originalverpackung des LEGO Mindstorms NXTs



Abb. 7: LEGO Mindstorms NXT mit allen Sensoren und Servomotoren

NXT Servomotoren

Die interaktiven Servomotoren enthalten ein aufwändiges Getriebe im Innern und haben bereits einen Rotationssensor eingebaut, welcher die relative Position des jeweiligen Servomotors zur Grundposition in Grad messen kann. Vier NXT Servomotoren bewegen die vier Arme des Delta-Roboters. Ein Nachteil der NXT Servomotoren ist, dass bei einer hohen Drehgeschwindigkeit die Rotationssensoren zunehmend ungenau werden. Da das erfolgreiche Zusammenbauen der Moleküle stark davon abhängt, ob die Arme eine präzise Bewegung machen, war das Arbeiten mit kleinen Drehgeschwindigkeiten erforderlich. Doch weil das Drehmoment bei kleinen Drehgeschwindigkeiten enorm schwach wird, waren die Servomotoren nicht mehr im Stande, die Arme zu bewegen. Dieses Problem wird weiter im Kapitel 4.4 “Probleme und Lösungen” besprochen.

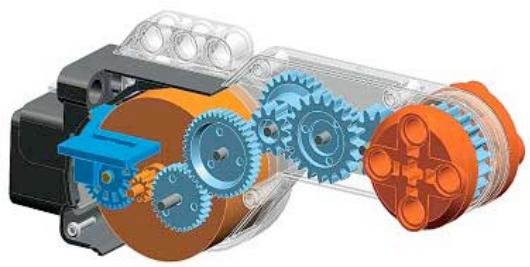


Abb. 8: Getriebe im Innern eines NXT Servomotors



Abb. 9: NXT Servomotor



Abb. 10: NXT Berührungssensor



Abb. 11: Kubischer Elektromotor

Kubischer Elektromotor

Dieser Motoren-Typ wurde für das Greifsystem verwendet, da er um einiges kleiner und leichter ist als der NXT Motor. Er besitzt keinen Rotationssensor und wird über ein altes LEGO-Elektronik-Kabel gespeist. Um vom NXT trotzdem angesteuert werden zu können, musste zusätzlich ein spezielles Verbindungskabel verwendet werden.



Abb. 12: Delta-Roboter Prototyp

Prototyp

Der erste, noch vor der Themenwahl konstruierte Delta-Roboter entsprach dem klassischen Modell. Er besaß drei Arme und für deren Steuerung drei Motoren. Bei der Konstruktion war schnell klar, dass der Prototyp durch die Anordnung der drei Motoren in 120°-Winkeln mit LEGO zwar möglich, doch zu instabil war und bei jeder Bewegung der Arme wackelte, da diese den Schwerpunkt des Roboters verlagerten.

Im Verlauf der Maturitätsarbeit entwickelte sich der Roboter von einem simplen, batteriebetriebenen, dreimotorigen, an einem Schreibtisch befestigten Prototypen, zu einem ausgereiften Roboter: Auf einer eigenen Plattform, an Aluminium-Trägern befestigt, steuern zwei, von einem Netzteil gespeiste NXT-Blöcke über fünf Motoren den vierarmigen Roboter mit Greifsystem. In den nachfolgenden Unterkapiteln wird dieser Entwicklungsprozess Schritt für Schritt beschrieben.

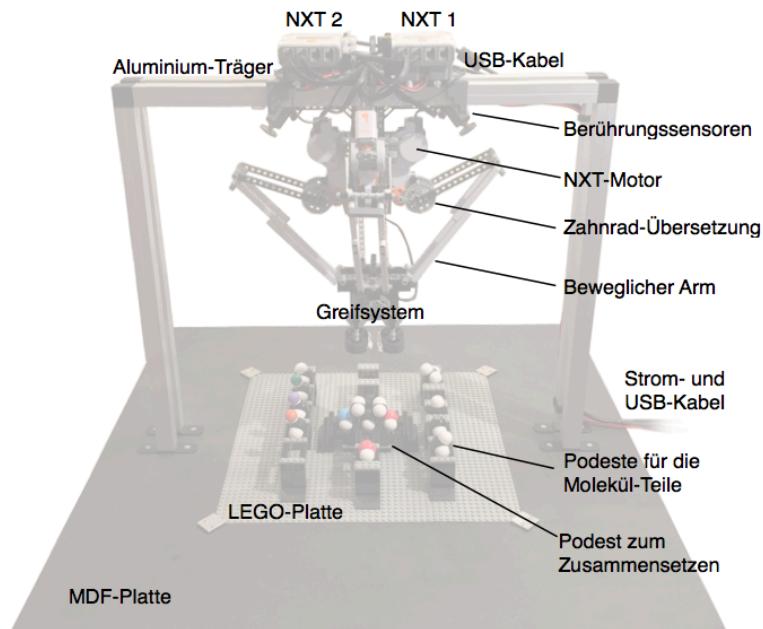


Abb. 13: Fertiger Roboter mit Beschreibung

Bauweise

Die zwei NXT-Einheiten wurden auf den Horizontalstangen der beiden Aluminium-Träger befestigt und unterseitig mit vier Schutzwulstern fixiert. Sowohl der zweifache Träger-Rahmen als auch verschiedene LEGO-Querverstrebungen verhindern ein übermässiges Wackeln und tragen zur Stabilisierung bei. Während dieses Prozesses hat der Roboter einige Versionen durchlaufen bis ein stabiles System für die Befestigung an den Aluminiumstangen gefunden wurde.

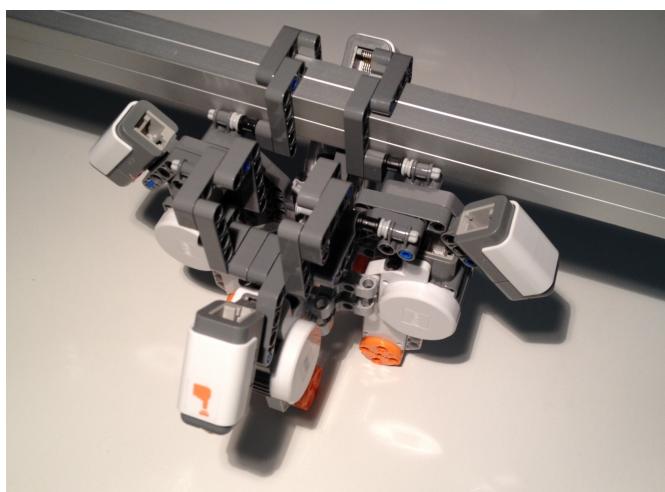


Abb. 14: Motorenaufhängung mit einer Aluminium Stange

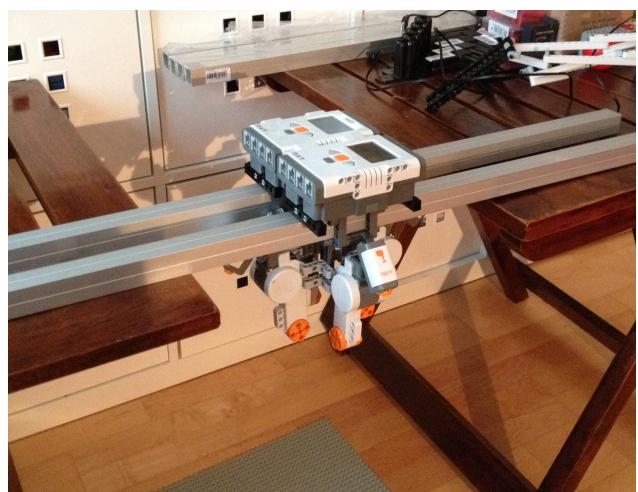


Abb. 15: Roboter an zwei Aluminium-Stangen

Durch die Verwendung von zwei NXTs ergab sich die Möglichkeit, maximal sechs Motoren zu steuern: Die Bauweise mit vier Servomotoren, bzw. vier Armen, angeordnet im 90°-Winkel, zeigte sich als stabilste Version. Ein fünfter Motor ist für das Greifsystem zuständig.

Die Funktionsweise der Arme ist die selbe, wie beim dreimotorigen Prototypen. Das obere Segment der Arme ist fix und am Motor (bzw. an der Übersetzung) befestigt, das untere Segment ist ein bewegliches Parallelogramm und mit dem Greifsystem verbunden. Ausgehend von den Armen konnten die Berührungssensoren so angebracht werden, dass sie von den Armen erst berührt werden, wenn sich diese in der Grundposition befinden, also maximal hochgeklappt sind.

Das Greifsystem des Roboters musste mehrere Anforderungen erfüllen: Es musste klein und zugleich leicht sein, damit es von den NXT Motoren der Arme bewegt werden konnte, es musste durch einen LEGO Motor steuerbar sein und als wichtigste Eigenschaft musste es die Molekulkugeln greifen, festhalten und wieder absetzen können. Es war viel Tüftelei nötig, bis das Greifsystem all diese Bedingungen erfüllen konnte. Die Problematik des Greifsystems wird im Unterkapitel 4.4 “Probleme und Lösungen” genau erklärt werden.

Schliesslich wurden die Arme an den Motoren und am Greifsystem angebracht und alle Motoren und Sensoren mit den NXTs verkabelt. Somit war der NXT für einen Testlauf bereit.

Für die finale Version des Roboters wurden noch möglichst viele farbige LEGO-Teile mit schwarzen, grauen oder weissen ersetzt, um ein einheitliches Aussehen zu erreichen.

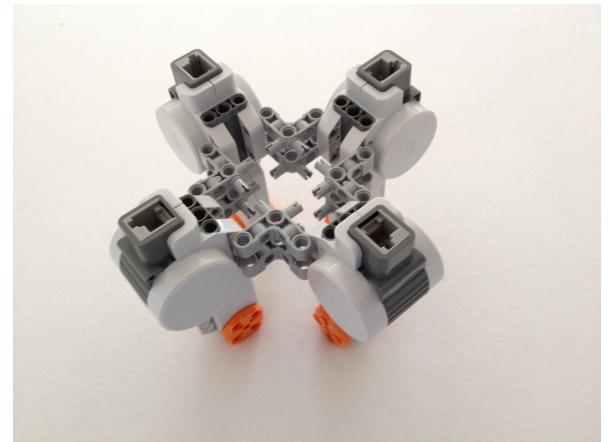


Abb. 16: Motorenaufhängung

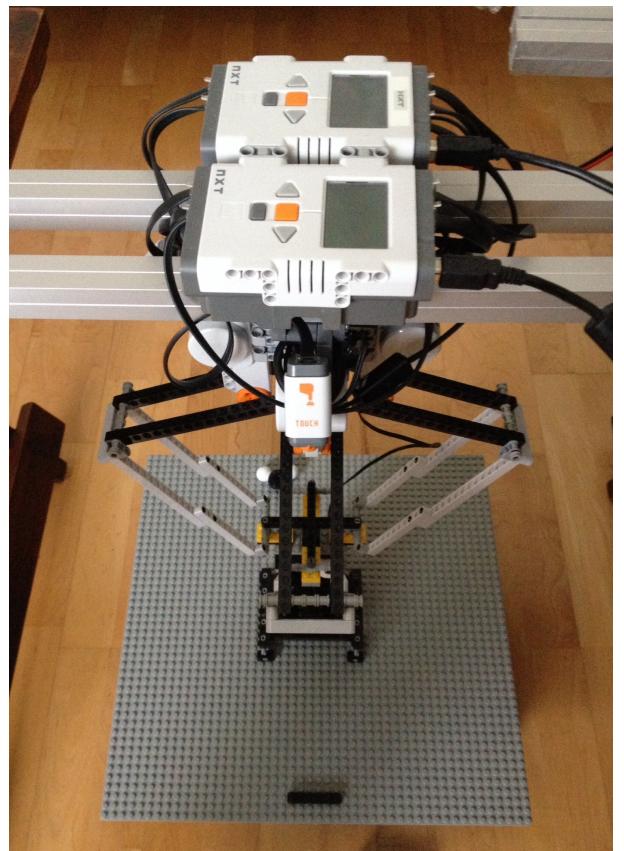


Abb. 17: Prototyp an zwei Aluminium-Stangen

4.3 Molekülsystem

Es musste ein Molekülmodell entwickelt werden, dessen Einzelteile für den Roboter greifbar und in verschiedenen Variationen zusammensetzbare sind.

Versuche mit Holz und Styropor

Die erste Idee bestand darin, Teile eines Kalottenmodells selber aus maschinell gefertigten Holzkugeln herzustellen und zu bemalen. Anfänglich betrug der Durchmesser der gewählten Kugeln fünf bis sechs Zentimeter. Bald zeigte sich aber, dass die massiven Holzkugeln für den Roboter zu schwer waren.

Styropor war die nächste Idee. Doch auch das Basteln mit diesen leichten Kugeln ergab kein saubereres, zufriedenstellendes Resultat.

Verwendung und Bearbeitung des Molymod-Molekülbaukastens

Nach weiterer Recherche zeigte sich der Molymod-Molekülbaukasten als beste Lösung: da zuvor das Molymod-System nur Stäbchenmodell bekannt war, war die neu entdeckte Möglichkeit des Kalottenmodells gerade ideal.

Ein Molymod-Molekülbaukasten wurde gekauft und das Greifsystem des Roboters für die kleinen Kugeln modifiziert.

Die Teile des Molymod-Systems sind aus solidem Kunststoff, und die Verbindungselemente halten hervorragend zusammen. Das Zusammensetzen der Teile ist aber bereits von Hand sehr hand relativ mühsam und erfordert viel Kraft. Für den Roboter war dies unmöglich auszuführen. Für die Verbindungsteile musste eine Alternative gefunden werden.

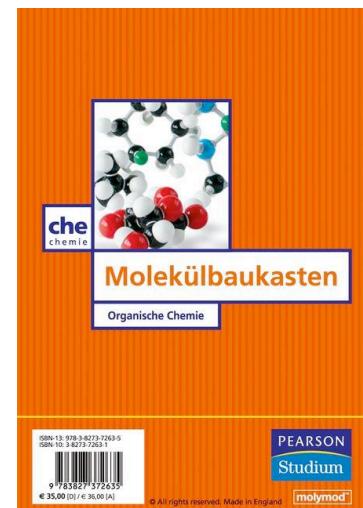


Abb. 18: Molymod-Molekülbaukasten

Verbindungs-Magnete

Um dieses Problem zu lösen, wurden Permanent-Magnete in die Löcher der Molymod-Atome gesteckt, die nun ganz einfach von selbst zusammenhielten und sich ohne grosse Mühe wieder trennen liessen. Vorgängig war ein zufälliger Versuch mit Magnet-Kugeln namens Neocube ausgesprochen erfolgreich. Die Neocube Magnet-Kugeln passten exakt in das Loch der Molymod-Atome. Es war kein Klebstoff notwendig. Außerdem weist das Magnet-Systems einen weiteren Vorteil auf: durch die magnetische Anziehungskraft werden kleine Abweichungen beim eventuellen Verrutschen oder Wegkippen eines Molekül-Teils von selbst korrigiert.



Abb. 19: Molekül-Teil mit Verbindungs-Magnet

Wahl und Anordnung der verschiedenen Molekül-Teile

Als Basis eignete sich eine grosse LEGO-Platte am besten. Mit kleinen, mit Holz unterlegten Aluminiumplättchen wurde diese auf der MDF-Plattform an jeder Ecke festgeschraubt. Nun galt es, einen passenden Aufbau zu konstruieren. Um die Anzahl und die Position der Molekül-Teile und die Vorrichtung, um sie zusammenzubauen, immer wieder abändern zu können, wurden die Molekülhalterungen (Podeste) ebenfalls aus LEGO gebaut. Dies stellte sich aber als grosse Schwierigkeit heraus. Auf die Problematik der Konstruktion der Podeste wird im Unterkapitel 4.4 “Probleme und Lösungen” eingegangen.

Zuerst sollte der Roboter mit einzelnen Atom-Kugeln arbeiten. Dies hätte zu viele Einzelteile benötigt und die baubaren Moleküle hätten deshalb nicht sehr komplex sein können. Weitere Versuche zeigten, dass es Sinn machte, bereits vorgebaute Molekül-Teile wie z.B. CH_2 zu verwenden: es können viel komplexere Moleküle gebaut werden können.

Zuerst wurden acht verschiedene Molekül-Teile um den Roboter in einem Kreis angeordnet. Daraus hätte der Roboter acht Molekülmodelle bauen können: Ethan bis Pentan und Ethanol bis Pentanol.

Als der Roboter jedoch zunehmend genauer wurde, konnte seine gesamte Reichweite ausgenutzt werden. Nun war die Konstruktion von länglichen Podesten aus LEGO möglich, auf denen schliesslich 14 Molekül-Teile platziert werden konnten.

Grundsätzlich braucht es sechs Molekül-Teile um einfache Kohlenwasserstoffketten (Ethan bis Pentan) zu bauen. Weitere acht unterschiedliche Molekül-Teile können an ein Ende einer Kohlenwasserstoffkette gesetzt werden:

- Doppelbindungen: $-\text{CH}-$ und CH_2 (planar)
- Amino-Gruppe: NH_2-
- Zwei Alkohol-Gruppen: HO- und $-\text{OH}$
- Halogene: ClCH_2- , ICH_2- , BrCH_2-

Diese 14 Molekül-Teile lassen sich zu 47 verschiedenen Molekülmodellen kombinieren. Ein Spezialfall ist das planare CH_2 Molekül-Teil, welches für Doppelbindungen verwendet wird. Damit es von beiden Magnetpolen angezogen werden konnte, wurde es anstatt mit einem Permanent-Magneten mit einem Reissnagel ausgestattet. Nun konnte dieses Molekül-Teil an beiden Seiten der Kohlenwasserstoff-Kette angebaut werden. Dabei zeigte sich, dass das Molekül-Teil leicht geneigt sein musste, damit es von den anderen Magneten angezogen werden konnte. Die dafür benötigte Neigung wurde durch das Anbringen hervorstehender LEGO-Teile erreicht.

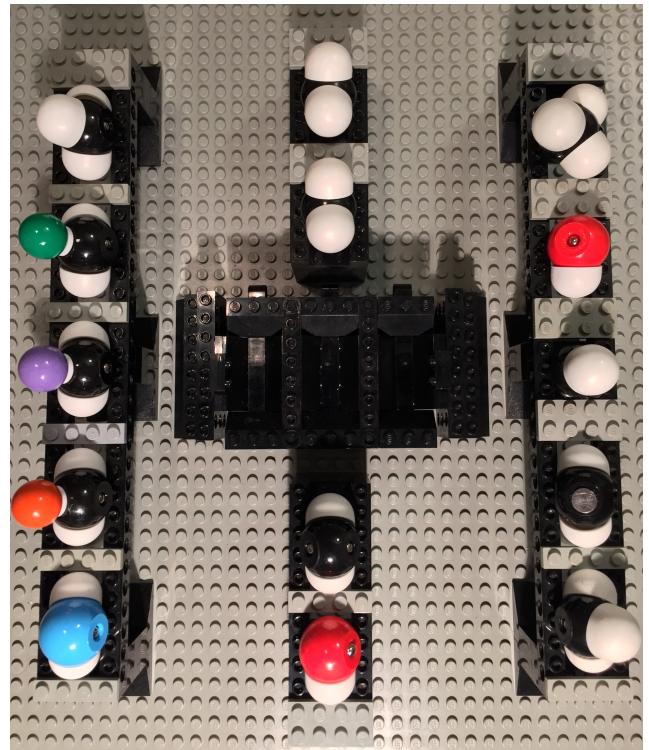


Abb. 20: Anordnung der Molekül-Teile

Wie das Programm mit den Positionen der einzelnen Molekül-Teilen umgeht, wird im Unterkapitel 5.5 “Erklärung des endgültigen Programms” genau beschrieben. Eine komplette Auflistung der baubaren Moleküle lässt sich im Anhang, Unterkapitel 14.2 “Tabelle der baubaren Moleküle” finden.

4.4 Probleme und Lösungen

In diesem Kapitel werden vier grössere Probleme genauer erläutert, welche bei der Konstruktion aufgetreten sind.

Versorgung der NXTs mit Netzteilen

Bereits nach einem Tag sporadischer Testläufe zeigte sich eine kontinuierliche Abschwächung der Motoren-Leistungen, und am Abend waren die zwölf verwendeten AA-Batterien leer. Mit aufladbaren Batterien verlief der Leistungsabfall noch schneller. Dies hatte zur Folge, dass die Bewegungen je nach Batteriestand zu sehr unterschiedlichen Ergebnissen führten, was die Leistungen des Roboters nicht vorhersehbar machte. Deshalb musste für die schwankende Batterieleistung dringend eine Lösung gefunden werden. Für einen stationären Roboter war eine Stromversorgung über ein Netzteil die beste Option. Dass dies schon bei anderen Leuten mit demselben Problem erfolgreich funktioniert hatte, bestätigte eine Recherche. Dafür wurden die sechs AA-Batterien zu je 1,5 Volt durch zwei 9 Volt-Netzteile zu 2 Ampere ersetzt.

Durch vier, aus Holz eigens dafür gefertigte, batterieförmige Zylinder, wurde je ein Kabelkanal gebohrt, die Kabel durchgezogen und die Enden der Kabel wiederum an Reissnägeln festgelötet. Zudem wurde auf den Minus-Pol ein Tropfen Lötzinn gegeben, um den Kontakt zu den Stromabnehmern der NXTs zu sichern.

Zuletzt wurden die Reissnägel auf die Holz-Batterien geklebt und diese wiederum in die Batteriefächer eingesetzt. Die Erleichterung war gross, als der Start-Ton der NXTs ertönte, nachdem die Netzteile an den Strom angeschlossen worden waren. Das Problem der schwankenden Energieversorgung war erfolgreich gelöst.

Präzision der Motoren

Obwohl die Motoren durch die konstante Stromversorgung durch die Netzteile keine Leistungsschwankungen mehr zeigten, war das Problem der Präzision der Motoren leider noch nicht gelöst. Da bei einem Delta-Roboter schon kleinste Ungenauigkeiten der Motoren grosse Auswirkungen auf die Position der Arme haben, wurde das Greifen eines Molekül-Teils beim Roboter zur Glückssache. Zwar sind die Motoren bei niedrigen Drehgeschwindigkeiten viel genauer, doch dann ist das Drehmoment der Motoren zu schwach, die Arme zu bewegen.



Abb. 21: Löten der Kabel

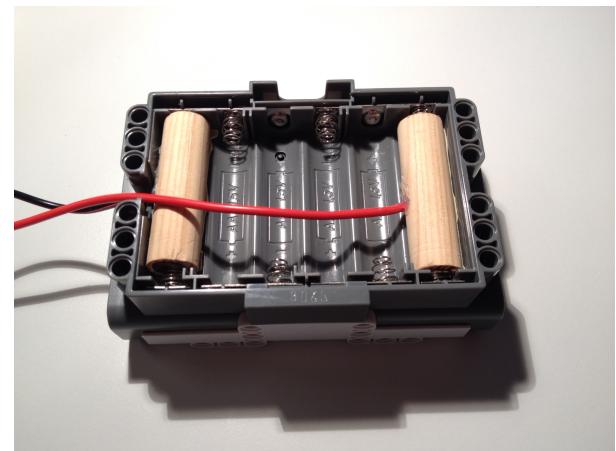


Abb. 22: NXT-Box mit Holz-Batterien

Für den Bau eines ganzen Moleküls ist also die Präzision der Motoren ausschlaggebend. Als Lösung wurde eine Zahnrad-Übersetzung für jeden Motor eingebaut: eine Kombination eines 8-zackigen Zahnrads, welches um die Achse der Motoren dreht, und eines 40-zackigen, welches an dem jeweiligen Arm befestigt ist. Somit muss sich der Motor fünfmal mehr drehen als zuvor, um die gleiche Winkelveränderung der Arme zu erreichen. Durch die neue Zahnrad-Übersetzung hatte sich zusätzlich auch die Drehrichtung geändert, weshalb alle Werte der Motoren-Winkel im Programm geändert werden mussten.

Dieses System ist viel toleranter, da sich die Präzisionsfehler der Motoren nur minimal auf die Position der Arme auswirken. Um zu schauen, wie sich die Übersetzung auf die Präzision des Roboters auswirkte, musste zuerst eine Aufhängung für die Zahnräder an die Motoren gebaut werden.

Nun stimmte aber der innere Abstand der Arme nicht mehr. Die Konstruktion der Arme musste nochmals überdenkt und umgebaut werden.

Doch der Aufwand hatte sich gelohnt: der Roboter bewegte sich nun mit einer erstaunlichen Präzision.

Greifsystem

Eine weitere Herausforderung war das Greifen der Molekül-Teile. Das Greifsystem musste eine gewisse Toleranz aufweisen, da die Aufgabe des Roboters darin bestand, viele verschiedene Molekül-Teile mit sehr unterschiedlichen Formen greifen zu können.

Verbindung zum NXT

Ganz zu Beginn des Projektes wurde ein NXT Servomotor als Greifmotor in Betracht gezogen, der vom NXT aus hätte gesteuert werden können. Doch aufgrund seiner umständlichen Form war es nicht möglich, ihn im Greifsystem zu integrieren. Die nächste Idee war eine mechanische Verbindung durch sich drehende Stangen, welche mit drehbaren Gelenken miteinander verbunden waren und deshalb die Bewegungen des Roboters mitmachen konnten. Doch aufgrund der sehr stark variierenden Winkel der Arme war diese Idee nicht umsetzbar. Die Lösung zeigte sich schliesslich in einem kleineren, leichteren und vor allem würfelförmigen LEGO-Motor. Durch ein spezielles Verlängerungskabel zum NXT stellten die Bewegungs-Abläufe des Delta-Roboters kein Problem mehr dar, und der Greifmotor konnte an jeder beliebigen Position erfolgreich vom NXT angesteuert werden.

Kralle

Die Konstruktion des ersten Greifsystems basierte auf einer Spielautomaten-Kralle. Dieses System eignete sich aufgrund der seitlichen Öffnungen im geschlossenen Zustand weder zum Greifen von kleinen, noch von grossen Objekten. Schnell wurde dieses System wieder verworfen.

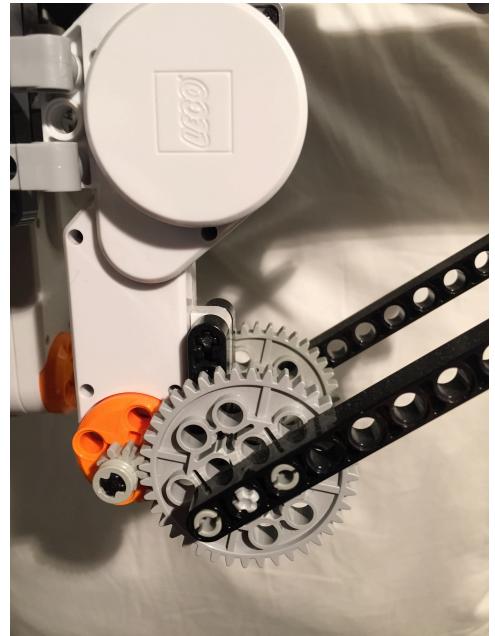


Abb. 23: Zahnrad-Übersetzung



Abb. 24: Kralle

Seitlich schliessendes Greifsystem

Ein seitlich schliessendes Greifsystem, ähnlich dem Öffnen und Schliessen einer Blende, zeigte sich als effektiver und war an einen Prototypen angebaut worden. Es bestand aus vier horizontal liegenden Reifen, die sich, jeder um eine eigene Achse, von vier Seiten kreisförmig nach innen bewegen. Als die Wahl jedoch auf das Molymod Molekülmodell fiel, musste auch dieses System verworfen werden, da es eher auf grössere Kugeln (Durchmesser ca. 5 cm) ausgelegt war. Ein weiterer Nachteil war, dass dieses System ganz gerade über einem Molekül-Teil heruntergefahren werden musste, um es nicht wegzustossen.

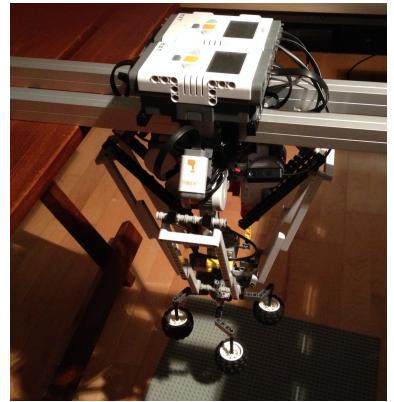


Abb. 25: Seitlich schliessendes Greifsystem

Sich von oben schliessendes Greifsystem

Ein weiteres Greifsystem bestand aus vier Stangen mit Reifen, die sich von oben herab um das Molekül kreuzweise schliessen und es festhalten. Leider hat dieses Greifsystem beim Praxistest völlig versagt, da die Molekül-Teile nicht von allen vier Reifen gleichzeitig festgehalten werden konnten und deshalb stets abrutschten.



Abb. 26: Sich von oben schliessendes Greifsystem

Zangenähnliches Greifsystem

Aufgrund der Form der Moleküle war schliesslich ein zangenähnliches Greifsystem konstruiert worden. Dieses konnte nun alle parallel auf der Plattform bereitliegenden Molekül-Teile greifen.

Zum Greifen der Moleküle wurden zuerst gerade, gummierte Stäbchen eingesetzt. Doch die Molekülkugeln rutschten einfach ab. Schliesslich wurden wieder vier Gummi-Reifen verwendet, an welche zusätzlich Isolationsröhrchen aus weichem Gummi angebracht wurden. Wenn sich das Greifsystem um ein Molekül-Teil schliesst, wird der Luftkanal im Röhrchen der Form entsprechend zusammengedrückt und das Molekül-Teil wird vom Gummi gehalten.

Doch erst nach mehreren Modifikationen des Greifsystems ergaben sich optimale Resultate, da der Abstand der Räder und das Drehmoment der Achsen mechanisch genau an die Molekül-Teile angepasst werden mussten.

Als letzte Variante wurde zur Probe ein Rotationssystem mit einem zweiten Motor eingebaut. Doch wegen seiner Grösse und seinem Gewicht wurde darauf verzichtet.

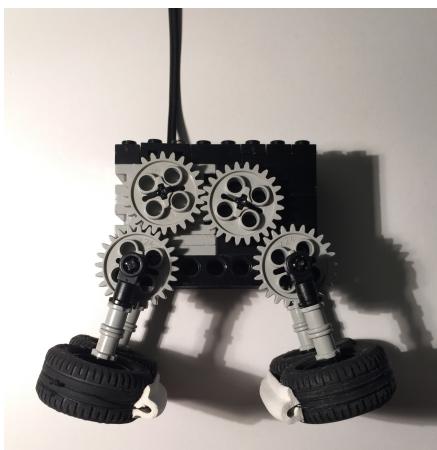
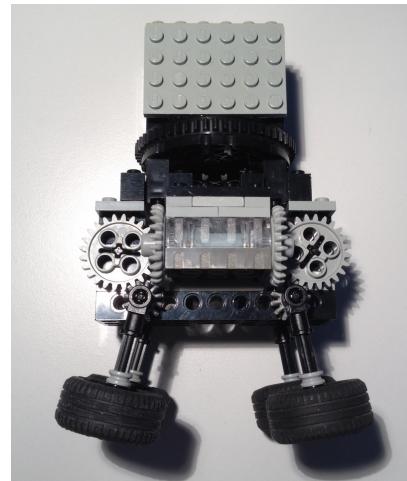


Abb. 27: Zangenähnliches Greifsystem



Seite 15

Abb. 28: Zangenähnliches Greifsystem mit Rotationssystem

Absetzen und Zusammensetzen der Molekül-Teile

Nun musste eine Vorrichtung gefunden werden, in welche der Roboter die einzelnen Molekül-Teile hineinlegen und sie so zusammenbauen konnte. Dies erwies sich als äusserst schwierig, da der Roboter die einzelnen Molekül-Teile aus einer Höhe von ca. eins bis zwei Zentimeter fallen lassen musste, um die bereits abgesetzten Molekül-Teile beim Öffnen des Greifsystems nicht wegzuschieben. Die erste Vorrichtung wurde aus LEGO gebaut, doch bereits beim Testen wurde klar, dass die Molekül-Teile an einem so harten Material wie LEGO einfach abprallen und sich deshalb nicht mehr mit den anderen Molekül-Teilen verbinden konnten. Ein weiteres Problem war, dass die einzelnen Molekül-Teile beim Absetzen kippten und für den Weiterbau des Moleküls nicht mehr in der richtigen Neigung waren. Darauf folgten mehrere Versuche mit Schaumstoff. Es wurden rechteckige Felder mit einem Teppichmesser aus einer Schaumstoffplatte herausgeschnitten, welche genau die breite der Molekül-Teile hatten. Das Hauptproblem bei Schaumstoff war, dass wenn der Roboter die Molekül-Teile nicht genau im ausgeschnittenen Feld abgesetzt hatte, diese dann vom Schaumstoff (wie von einem Trampolin) abprallten und nicht im ausgeschnittenen Feld liegen blieben. Eine weitere Vorrichtung wurde aus Trockensteckschaum für Blumen gefertigt, eine unbefriedigende Lösung.

Glücklicherweise wurden rampenförmige LEGO-Teile entdeckt. Sie erlaubten eine saubere und von der Neigung her ideale Form. Die Molekül-Teile werden bei dieser Vorrichtung von allen Seiten von den geneigten Flächen gegen die Mitte geleitet. Diese Lösung wurde genau unter dem Roboter auf der LEGO-Platte aufgebaut.

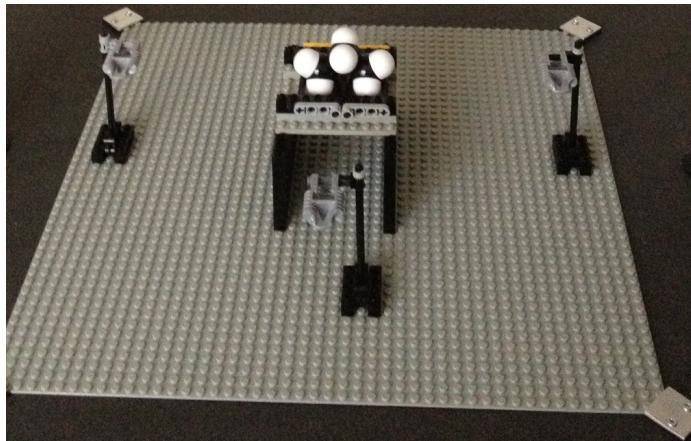


Abb. 29: Absetz-Vorrichtung aus LEGO (Prototyp)

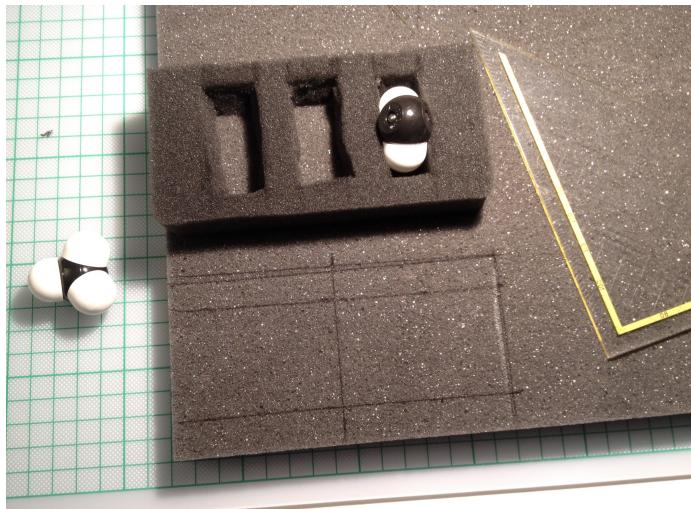


Abb. 30: Absetz-Vorrichtung aus Schaumstoff

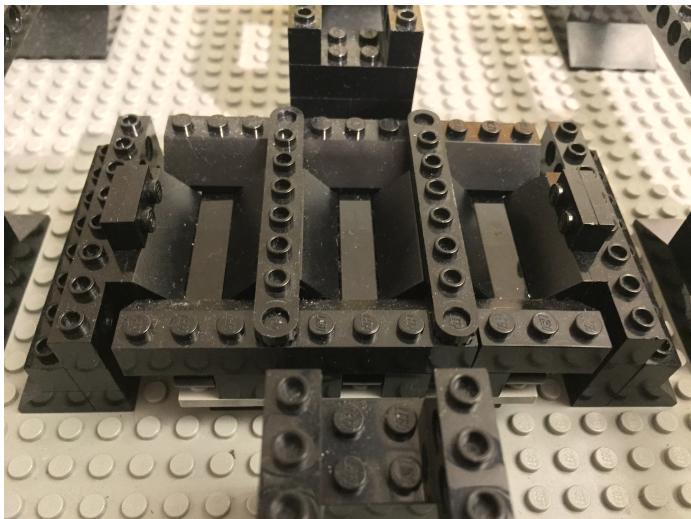


Abb. 31: Absetz-Vorrichtung aus LEGO (definitiv)

5. Programmierung der Software mit LabVIEW

5.1 Einleitung

LabVIEW ist eine graphische Entwicklerplattform der Firma National Instruments, welche von Ingenieuren und Hochschulen auf der ganzen Welt verwendet wird. Im Projekt wurde LabVIEW benutzt, um den Roboter zu programmieren und ihn durch ein GUI zu kontrollieren.

Glücklicherweise habe ich beim 24h-Wettbewerb 2014 von Helveticrobot mit Herrn Vance Carter kennengelernt, den Geschäftsführer der EducaTec AG.

Als ich ihm von meinem Projekt mit LEGO Mindstorms NXT berichtete, hatte er nicht nur LabVIEW empfohlen, sondern mir danach auch sogar die Installations-CD für LabVIEW 2012 for Education ausgeliehen, auf der die ganzen Datenbanken für die Interaktion mit dem LEGO Mindstorms NXT (kurz: LabVIEW for LEGO Mindstorms) enthalten waren.

In den Sportferien 2014 habe ich dann begonnen, mich mit LabVIEW zu befassen, auf YouTube Tutorials über LabVIEW geschaut und mit LabVIEW und dem NXT zu experimentieren, um mir ein Grundwissen für die Maturitätsarbeit anzueignen.

5.2 LabVIEW als graphische Programmiersprache

Bis dahin kannte ich nur die originale graphische Programmiersprache NXT-G von LEGO. Diese basiert in vereinfachter Weise auf LabVIEW. Im Gegensatz zu der NXT-G Programmiersprache hat man in LabVIEW zwei Möglichkeiten einen NXT zu programmieren:

1. Das Programm wird auf den NXT-Block heruntergeladen. Der Roboter funktioniert jetzt autonom und befolgt den Algorithmus des Programms.
2. Der NXT ist die ganze Zeit mit dem Computer entweder über USB oder über Bluetooth verbunden. Das Programm läuft auf dem Computer und gibt dem NXT Anweisungen, z.B. die Motoren zu drehen oder einen Ton abzuspielen. Der NXT wiederum versorgt das Programm mit Sensordaten, welche der Computer dann weiterverarbeiten kann.

Für die Programmierung des Roboters wurde die zweite Variante ausgewählt, da ein GUI auf dem Computer verwendet werden sollte.

In LabVIEW hat jedes einzelne Programm (oder Virtual Instrument wie es in LabVIEW genannt wird) zwei Fenster: Ein Front Panel und ein Block Diagramm:

Auf dem Frontpanel kann eine Benutzeroberfläche mit Tasten, Schaltern, Reglern, LEDs, Anzeigen, Textboxen, Tabellen, Graphiken und vielen weiteren Objekten erstellt werden. Jedes Objekt, welches einen Input (z.B. ein Regler) oder einen Output (z.B. eine Anzeige) hat, wird auf dem Block Diagramm als Block dargestellt.

Auf dem Block Diagramm erfolgt die eigentliche Programmierung. In LabVIEW werden einzelne Aktionen im Algorithmus als Blöcke dargestellt. Die Blöcke haben einen oder mehrere In- und Outputs. Auf dem Blockdiagramm können diese Blöcke miteinander verbunden werden. Die Verbindungen sind verschiedene Datenleitungen, welche Informationen von Block zu Block transportieren. Die Informationen können z.B. Zahlen, Strings (also Text) oder eine Boolesche Variable (True oder False) sein. Des Weiteren können die Blöcke in verschiedenen Kästchen zusammengesetzt werden, welche den Ablauf des Programms beeinflussen. Normalerweise werden die Daten von links nach rechts transportiert, weshalb sich die Inputs eines Blocks meistens links und die Outputs rechts befinden.

Für die Programmierung eines NXT Roboters wird eine pinkfarbene Datenleitung verwendet. Alle Blöcke, welche mit dem NXT interagieren, besitzen sowohl einen In- als auch einen Output für diese Leitung. Um einen Algorithmus zu programmieren, werden die verschiedenen Blöcke in Reihen miteinander verbunden, welche vom Programm von links nach rechts gelesen werden. Dies macht die Programmierung übersichtlich und kompakt.

Ein weiterer grosser Vorteil von LabVIEW ist, dass mehrere Prozesse gleichzeitig ablaufen können. Somit können z.B. während des ganzen Programms die Winkel der Motoren gemessen und angezeigt werden.

5.3 Die wichtigsten Elemente von LabVIEW für das Programm

In diesem Unterkapitel werden die wichtigsten verwendeten Elemente von LabVIEW kurz erklärt.

Motor On Block

Der Motor On kann einen oder mehrere Motoren eines NXTs einschalten und mit einer bestimmten Geschwindigkeit drehen lassen. Er hat einen Input, mit welchem die Ausgangsports ausgewählt werden können, die der Block dann aktiviert, und einen weiteren namens Power, welcher für die Drehgeschwindigkeit verantwortlich ist. Dieser kann einen Wert zwischen 100 (maximale Drehgeschwindigkeit vorwärts) und -100 (maximale Drehgeschwindigkeit rückwärts) haben. Links und rechts hat der Block je einen In- und Output, welche für die NXT Datenleitung dienen, um die Reihenfolge des Algorithmus festzulegen.

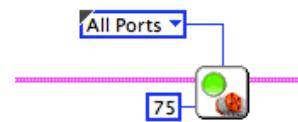


Abb. 32: Motor On Block

Motor Coast und Motor Brake Blöcke

Der Motor Coast und der Motor Brake Block stoppen die Bewegung eines Motors auf verschiedene Weisen: Der Motor Coast Block unterbricht einfach die Stromzufuhr und lässt den Motor austrudeln. Der Motor Brake Block bremst den Motor und hält ihn fix in der jeweiligen Position. Beide Blöcke haben einen Input, mit welchem die betroffenen Ausgangsports ausgewählt werden können und einen In- und Output für die NXT Datenleitung.

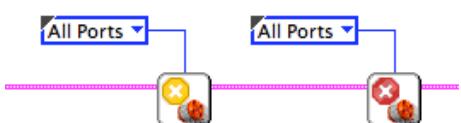


Abb. 33: Motor Coast und Motor Break Block

Wait For Pressed Block

Der Wait For Pressed Block verhindert, dass der Algorithmus weiter abläuft bis ein Berührungssensor gedrückt wird. Der Block hat einen Input, über welchen der Eingangsport ausgewählt werden kann und auch wiederum einen In- und Output für die NXT Datenleitung.

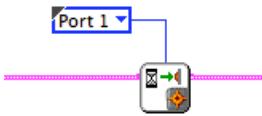


Abb. 34: Wait For Pressed Block

Reset Rotation und Wait For Rotation Block

Der Rotationssensor misst die Rotation eines Motors in Grad. Dreht der Motor eine ganze Umdrehung vorwärts, misst der Sensor 360° . Dreht er stattdessen eine ganze Umdrehung rückwärts, werden -360° gemessen. Der Reset Rotation Block setzt den gemessenen Wert des entsprechenden Rotationssensors auf 0° zurück. Der Wait for Rotation Block verhindert, dass der Algorithmus weiterläuft, bis der Motor einen gewissen Winkel erreicht. Beide Blöcke haben einen Input um die betroffenen Eingangsports auszuwählen und je einen In- und Output für die NXT Datenleitung.



Abb. 35: Reset Rotation und Wait For Rotation Block

Specify NXT Block

Der Specify NXT Block ist der Anfangsblock des Programms. Er wird nur gebraucht, falls das Programm mehrere NXTs ansteuern muss. Er hat je einen Input für den Namen des NXTs und einen für den Verbindungstyp (USB oder Bluetooth). Dieser Block startet den Algorithmus. Er ist über den Output mit einer NXT Datenleitung verbunden.



Abb. 36: Specify NXT Block

Tone Block

Der Tone Block spielt über den NXT einen Ton ab. Durch zwei Inputs können sowohl die Tonhöhe, als auch die Dauer des Tones bestimmt werden. Er dient als Orientierungshilfe während des Programm-Ablaufs. Zusätzlich hat dieser Block auch noch je einen In- und Output für die NXT Datenleitung.

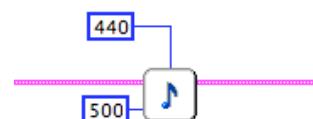


Abb. 37: Tone Block

Textinput

Dieses Objekt existiert einmal auf dem Front Panel als Textbox und einmal auf dem Block Diagramm als Block mit einem Output. Der eingegebene Text kann als String-Datenleitung im Block Diagramm verarbeitet werden.

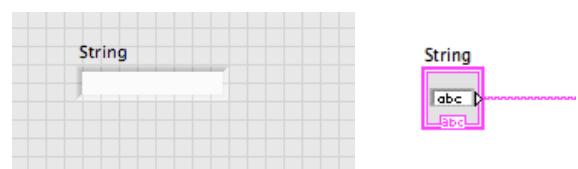


Abb. 38: Textinput

Taste

Dieses Objekt existiert einmal auf dem Front Panel als Taste und einmal auf dem Block Diagramm als Block mit einem Output. Der Output hat einen Booleschen Wert, welcher zeigt, ob die Taste gedrückt ist (True) oder nicht (False).

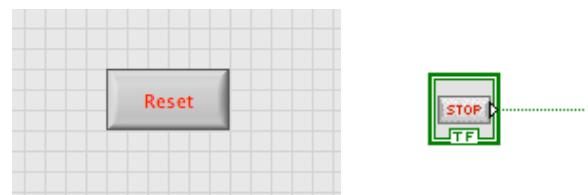


Abb. 39: Taste

LED

Dieses Objekt existiert einmal auf dem Front Panel als LED und einmal auf dem Block Diagramm als Block mit einem Input. Die LED zeigt auf dem Front Panel an, welchen Booleschen Wert der Input im Block Diagramm erhalten hat. Bei True leuchtet die LED auf dem Front Panel, bei False ist sie ausgeschaltet.

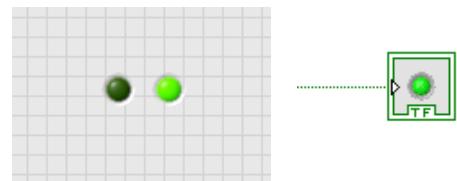


Abb. 40: LED

While Loop

Der While Loop ist in LabVIEW ein Kästchen, welches um andere Blöcke herum platziert werden kann. Der While Loop wiederholt die Programmanweisungen im Kästchen so lange, bis die Loop Condition den Loop beendet (ausgelöst durch einen Booleschen Wert). Zusätzlich besitzt der While Loop einen Output für die Iteration (Wiederholungs-Index), welcher bei 0 beginnt und sich bei jeder Iteration um eins vergrößert.



Abb. 41: While Loop

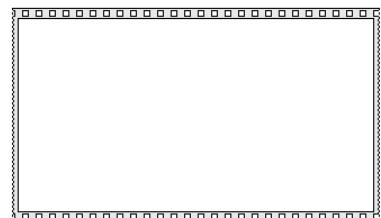


Abb. 42: Flat Sequence Structure

Flat Sequence Structure

Die Flat Sequence Structure ist ebenfalls ein Kästchen und kann um andere Blöcke herum platziert werden. Sie bewirkt, dass zuerst alle Programmanweisungen im Kästchen ausgeführt werden müssen, bevor das Programm weiterläuft kann.

Case Structure

Die Case Structure entspricht dem If...Then in textbasierten Programmiersprachen. Es ist ein Kästchen mit einem Dropdown-Menü welches alle möglichen Fälle aufzeigt. Es hat einen Input, welcher z.B. eine Zahl, ein Boolescher Wert oder ein String sein kann. Dieser Input bestimmt dann welcher Fall zutrifft und löst die entsprechende Programmanweisung aus.

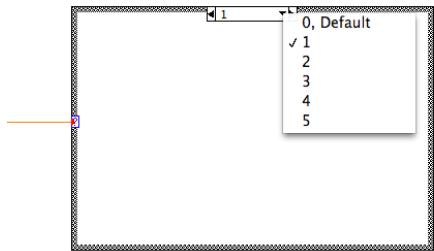


Abb. 43: Case Structure

5.4 Übungsprogramme

Um sich mit LabVIEW langsam an das gewünschte Programm heranzutasten, wurden eine Reihe von Übungsprogrammen erstellt, welche mit dem jeweiligen Prototypen des Delta-Roboters getestet wurden.

Begonnen wurde mit der Eichung der Rotationssensoren. Die Motoren drehten die Arme langsam nach oben, bis die einzelnen Arme den jeweiligen Berührungssensor berührten und dann abbremsen. Als letzter Schritt wird der Rotationssensor-Wert zurückgesetzt. Somit ist die Ausgangsposition der Arme jedes Mal genau die selbe, und der Roboter kann seine Rotationssensoren eigenständig eichen.

Es folgte ein Programm, welches sich zuerst von selbst eicht, dann eine simple, seitliche Bewegung der Arme auslöst und sich schliesslich wieder eicht. Zusätzlich wurde das Greifsystem so programmiert, dass es sich schliesst sobald die seitliche Bewegung abgeschlossen ist.

Als nächstes wurde ein erstes GUI erstellt mit acht, in einem Kreis angeordneten, virtuellen Tasten. Wird eine Taste gedrückt, bewegt der Roboter das Greifsystem zu diesem Punkt und lässt es schliessen.

Die Werte der Motoren-Winkel für jede Bewegung wurden in Form von fixen Tabellen, sogenannte Arrays, eingegeben und abgespeichert. Das Tasten-System wurde wiederum durch ein Textfeld ersetzt, das verschiedene Bewegungen auslösen konnte.

Es wurden auch noch einige LEDs hinzugefügt, um zu signalisieren, ob der Roboter am bauen ist. Aus diesem relativ einfachen Programm entwickelte sich nach und nach das endgültige und voll funktionstüchtige Programm, welches im nächsten Unterkapitel genau erklärt wird.

5.5 Erklärung des endgültigen Programms

Um das Programm klar erklären zu können, müssen zuerst das System der Positionen der einzelnen Molekül-Teile und das GUI erläutert werden.

Die Positionen der Molekül-Teile

Die einzelnen Molekül-Teile mussten so unter dem Roboter platziert werden, dass sie sich einerseits in Reichweite des Greifsystems befanden und andererseits seitlich genug Abstand zueinander hatten, damit das Greifsystem einzelne Molekül-Teile greifen konnte, ohne die anderen zu berühren.

Die erste Idee war, acht Molekül-Teile in einem Kreis um den Roboter zu positionieren. Die Erhöhung der Präzision durch die Zahnrad-Übersetzung erlaubte nun die Anwendung von 14 Molekülteilen. Im Programm hat jede Position eine eigene Nummer von eins bis 14, welche die entsprechende Bewegung des Roboters auslöst. Zusätzlich gibt es fünf Positionen, mit den Nummern 15 bis 19, über welche der Roboter die Molekül-Teile absetzen kann.

Beim Testen der baubaren Moleküle war das Problem aufgetreten, dass gewisse Molekül-Teile beim Absetzen kippten.

Um dies zu beheben, wurden neue Positionen einprogrammiert, welche die problematischen Molekül-Teile je nachdem ein bisschen anders fallen ließen, damit sie nicht kippen konnten. Für die Positionen 15, 16 und 18 wurden deshalb die Spezial-Fälle 20, 21 und 22 einprogrammiert.

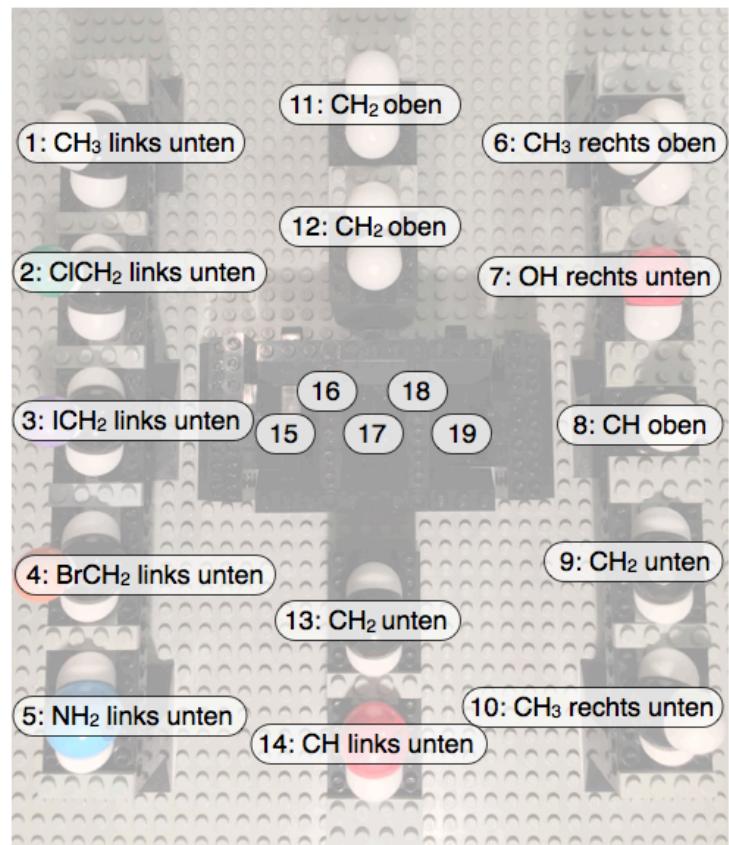


Abb. 44: Positionen der Molkekül-Teile mit Beschreibungen

Das GUI

Das GUI befindet sich auf dem Front Panel des LabVIEW Programms. Der Benutzer kann darauf das gewünschte Molekül auswählen und das Programm zeigt dann Informationen über den laufenden Bauprozess an. Da das GUI und der Roboter später in einem Film gezeigt und veröffentlicht werden sollen, wurde das GUI mit englischen Angaben versehen. Das GUI für das angewendete Programm weist folgende Funktionen auf:

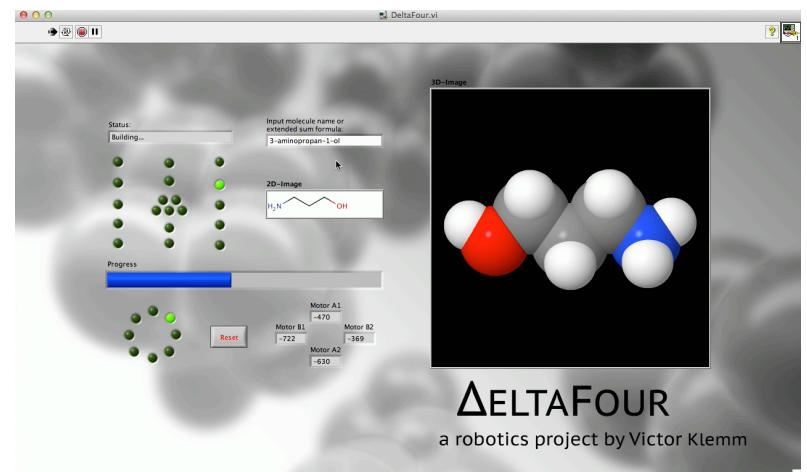


Abb. 45: GUI

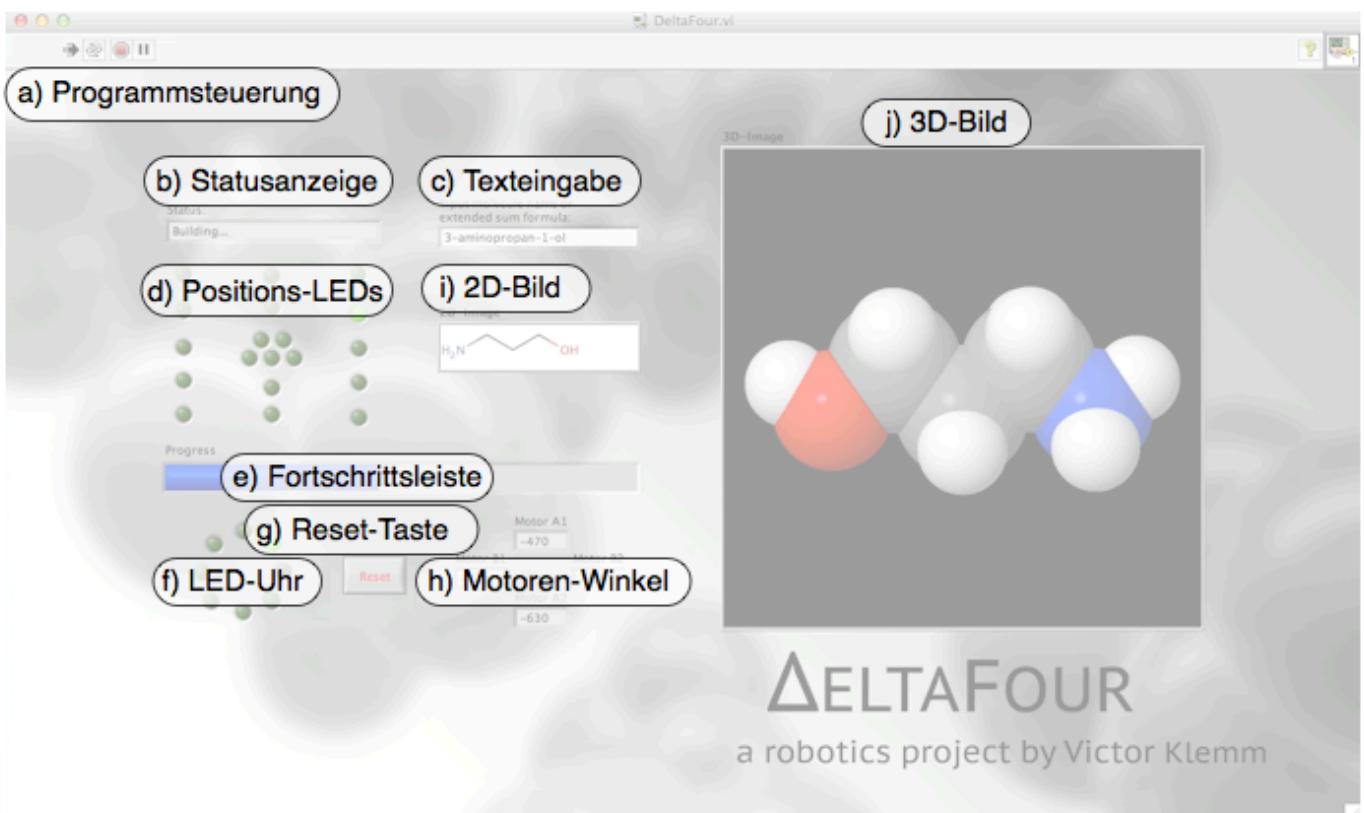


Abb. 46: GUI mit benannten Funktionen

a) Programmsteuerung

Die Programmsteuerung ist von LabVIEW vorgegeben. Mit ihr kann das Programm gestartet, bzw. gestoppt, pausiert oder wiederholt werden.

b) Statusanzeige

Die Statusanzeige gibt dem Benutzer ein Text-Feedback zum aktuellen Status des Programms:

Statusnachricht:	Wird angezeigt, sobald:
“Initialising...”	das Programm gestartet wurde
“Waiting for text-input”	der Roboter und das Programm für den Bauprozess bereit sind
“Building...”	ein gültiger Molekülname eingegeben wurde und der Bauprozess startet
“Not available”	ein ungültiger Molekülname eingegeben wurde
“Finished!”	der Bauprozess abgeschlossen ist
“Stop”	die Reset-Taste gedrückt wurde

c) Texteingabe

Mit der Texteingabe kann der Benutzer das gewünschte Molekül auswählen. Er kann das Molekül mit dem deutschen oder englischen Namen, oder mit der erweiterten Summenformel auslösen (Bei grösseren Molekülen funktioniert auch noch eine verkürzte Version der erweiterten Summenformel). Eine Übersicht der baubaren Moleküle und Namen befindet sich im Anhang, Unterkapitel 14.2 “Tabelle der baubaren Moleküle”.

d) Positions-LEDs

Die Positions-LEDs zeigen immer die aktuelle Zielposition des Roboters an. Deshalb gibt es analog zum Modell für jedes Molekül-Teil und jeden Platz zum Absetzen eine eigene LED. (Die Spezial-Fälle 20, 21 und 22 werden gleich angezeigt wie ihre Ursprungsfälle 15, 16 und 18.)

e) Fortschrittsleiste

Die Fortschrittsleiste füllt sich schrittweise während des Bauprozesses auf. Sie zeigt dem Benutzer graphisch, wie weit der jeweilige Bauprozess fortgeschritten ist.

f) LED-Uhr

Die LED's der LED-Uhr blinken nacheinander kreisförmig, um zu signalisieren, dass der Bauprozess läuft. Sobald dieser abgeschlossen ist, blinken alle LED's gleichzeitig.

g) Reset-Taste

Sobald der Bauprozess abgeschlossen ist, kann das Programm durch das Drücken der Reset-Taste gestoppt und das GUI zurückgesetzt werden.

h) Motoren-Winkel

Während des ganzen Programm-Ablaufs werden ständig die Winkel aller vier Servomotoren in Grad gemessen und auf dem GUI angezeigt.

i) 2D-Bild

Auf einem 2D-Bild wird die Skelettformel des ausgewählten Moleküls dargestellt.

j) 3D-Bild

Auf einem 3D-Bild wird das Kalottenmodell des ausgewählten Moleküls dargestellt.

Grobe Struktur des Programms

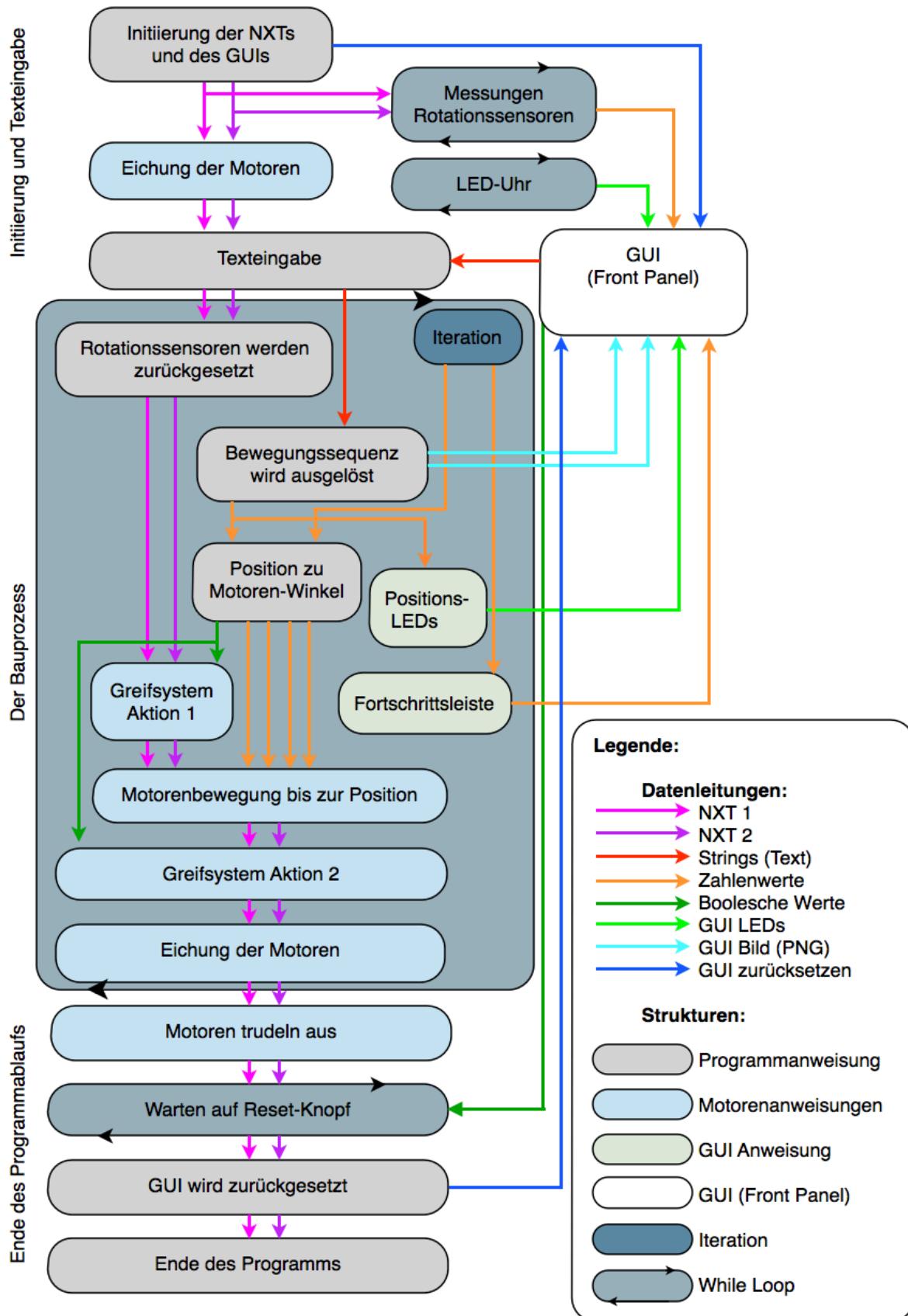


Abb. 47: Grobe Struktur des Programms

Initiierung und Texteingabe

Wenn das Programm gestartet wird, werden als erstes die beiden NXTs initiiert. Sobald diese bereit sind, ertönt ein Signalton. Ab diesem Moment werden die Messungen der Rotationssensoren auf dem GUI angezeigt. Als nächstes eicht der Roboter die vier Rotationssensoren der Servomotoren, indem er sie solange rückwärts drehen lässt, bis die Arme die Berührungssensoren berühren.

Nun ist der Roboter für den Bauprozess bereit und die Texteingabe aus dem GUI kann erfolgen. Bis eine gültige Texteingabe erfolgt ist, blinkt ein Lämpchen am Roboter, um zu signalisieren, dass eine Interaktion mit der Software benötigt wird. Sobald ein gültiges Molekül gewählt worden ist, wird der Bauprozess ausgelöst. Während des ganzen Bauprozesses blinkt die LED-Uhr, um zu signalisieren, dass ein Molekül gerade gebaut wird.

Bauprozess

Als erster Schritt werden die Rotationssensoren zurückgesetzt. Dann wird der Text-Input des Benutzers einem der 47 baubaren Molekül-Fälle zugeordnet und die entsprechende Bewegungs-Sequenz ausgelöst.

Im Programm ist eine Bewegungs-Sequenz eine Liste der Bewegungen (Greif-Positionen), durch die das Molekül zusammengebaut wird. Eine komplette Auflistung der Bewegungs-Sequenzen lässt sich im Anhang, Unterkapitel 14.2, “Tabelle der baubaren Moleküle” finden.

Neben der Bewegungs-Sequenz werden auch das entsprechende 2D- und das 3D-Bild ausgelöst und auf dem GUI dargestellt.

Als nächstes wird ein Element der Bewegungs-Sequenz in die Werte der vier Motoren-Winkel umgewandelt, (das Element wird durch die aktuelle While Loop Iteration bestimmt). Zusätzlich wird ein Boolescher Wert herausgegeben, welcher anzeigt, ob das Greifsystem ein Molekül-Teil greifen (Positionen 1-14) oder absetzen (Positionen 15-22) soll. Der Wert in der Bewegungs-Sequenz schaltet auch noch die entsprechende Positions-LED auf dem GUI ein.

Abhängig von den Werten werden nun die Motoren aktiviert. Als erstes öffnet sich das Greifsystem (greifen) oder es bleibt geschlossen (absetzen). Dann bewegen sich die Servomotoren, bis sie die Arme die gewünschten Winkel-Werte erreicht haben. Darauf schliesst sich das Greifsystem, um ein Molekül-Teil zu greifen, oder öffnet sich langsam nur wenig, um es abzusetzen. Dieses Öffnen des Greifsystems verhindert, dass bereits platzierte Moleküle aus ihrer Position verschoben werden. Am Ende der Bewegungssequenz drehen die Motoren wieder so lange, bis die Motoren in der Grundposition die Berührungssensoren berühren und Rotationssensoren wieder geeicht sind.

Dieser ganze Bauprozess wird solange wiederholt, bis der letzte Schritt der Bewegungssequenz erreicht wurde. Bei jeder Wiederholung lädt sich die Fortschrittsleiste auf dem GUI ein bisschen weiter auf. Am Ende der Bewegungssequenz steht eine 0, um dem Programm zu signalisieren, dass das Molekül fertig gebaut ist. Wenn dies der Fall ist, wird der While Loop der Bewegungssequenz gestoppt.

Ende des Programmablaufs

Darauf trudeln die Motoren aus. Das Programm wartet nun auf das Drücken der Reset-Taste. Währenddessen werden auf dem GUI immer noch die Bilder angezeigt und das Lämpchen am Roboter blinkt, um wieder zu signalisieren, dass eine Interaktion mit der Software benötigt wird. Sobald die Reset-Taste gedrückt wird, werden die Bilder und die Texteingabe auf dem GUI zurückgesetzt und das Programm beendet.

Das Programm ist nun bereit für eine neue Anwendung.

Wichtigste Elemente

In diesem Unterkapitel werden drei wichtige Elemente des Programms genauer beschrieben.

Eichung

Bei der Eichung drehen alle vier Servomotoren gleichzeitig solange, bis jeder Arm den jeweiligen Berührungssensor berührt. Ein Signalton ertönt und der Motor stoppt. Mit dieser Programmanweisung kann der Roboter immer wieder in die gleiche Grundposition zurückkehren, um durch die erneute Eichung die Präzision zu maximieren.

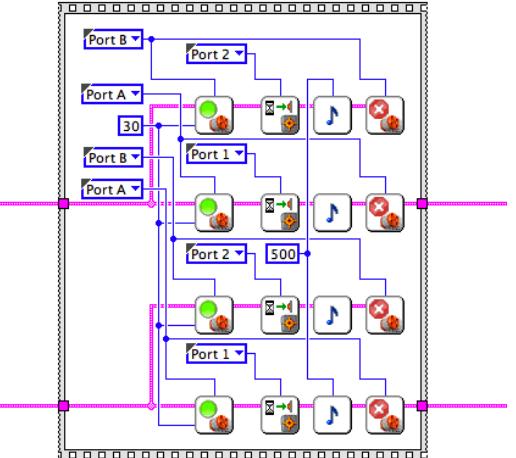


Abb. 48: Eichung

Texteingabe

Die Texteingabe findet innerhalb eines While Loops statt, welcher jede halbe Sekunde wiederholt wird. Der eingegebene Text löst innerhalb einer Case Structure eine Rückmeldung auf der Statusanzeige aus. Wenn eines der 47 gültigen Moleküle als einen Fall der Case Structure erkannt worden ist, endet der While Loop und das Programm läuft weiter.

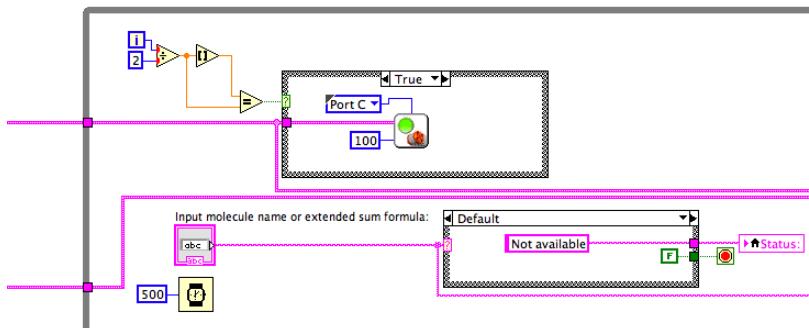


Abb. 49: Texteingabe

Bewegungskoordination

Bei dieser Programmanweisung drehen alle vier Servomotoren gleichzeitig so, dass sich die beweglichen Arme nach unten bewegen. Wenn ein Motor den gewünschten Motoren-Winkel erreicht hat, bremst er. Sobald alle Motoren gebremst haben, fährt das Programm weiter.

Diese Programmanweisung macht nur Sinn, wenn sich die Motoren anfangs in der Grundposition befunden haben und die Rotationssensoren geeicht worden sind.

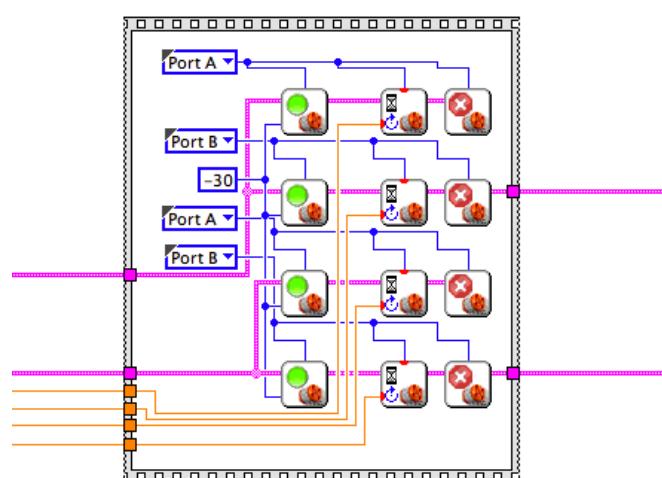


Abb. 50: Bewegungskordination

5.6 Anpassung des Programms auf die Molekülpositionen

Damit der Roboter die einzelnen Molekül-Teile exakt greifen und absetzen konnte, mussten die Winkel für alle vier Servomotoren genau auf die jeweilige Position abgestimmt werden.

Um die Winkel für eine Position zu berechnen, wurde mit sehr viel Mühe und Trigonometrie eine Funktion gefunden, welche die Winkel der Motoren für eine rein horizontale Bewegung berechnet.

Leider drehten die Motoren weiter als bis zum gewünschten Winkel, da das Programm Zeit brauchte, um zu reagieren und die Motoren zu bremsen.

Weitere komplizierte Berechnungen sämtlicher Winkel hatten sich daher erübrigten.

Die Werte mussten nun empirisch ermittelt werden. Ein extra dafür geschriebenes Programm half, diesen Prozess wesentlich zu vereinfachen: es zeichnete alle Motoren-Winkel der Servomotoren auf, während der Roboter von Hand zur gewünschten Position bewegt wurde. Die somit erlangten Näherungswerte mussten in kleinen Schritten getestet und nachkorrigiert werden, um schliesslich die bestmöglichen Winkel-Werte zu bestimmen. Dieses Verfahren war sehr zeitaufwendig und musste für jede Roboter-Version neu durchgeführt werden.

Das gleiche Verfahren wurde auch für die Positionen des Absetzens der einzelnen Molekül-Teile verwendet.

6. Diskussion

Eine Testserie hat ergeben, dass der Roboter jedes der 47 Moleküle erfolgreich bauen konnte. Der Roboter kann jedes der 14 Molekül-Teile selbstständig greifen, transportieren und wieder absetzen. Das Programm funktioniert wie gewünscht und das Molekül kann plangemäss durch das GUI ausgewählt werden.

Trotzdem kann das erfolgreiche Bauen gewisser Moleküle stark von den Anfangsneigungen der einzelnen Molekül-Teile abhängen, weshalb diese zuerst sorgfältig platziert werden müssen. Dass aufgrund einer Zeitverzögerung beim Computer oder bei den NXTs die Motoren nicht rechtzeitig bremsen und deshalb das Greifsystem das Molekül-Teil nicht am richtigen Ort greift oder loslässt, kann auch zu Fehlleistung führen. Glücklicherweise passiert dies sehr selten und kann leider nicht behoben werden.

Doch das Anfangsziel wurde klar erreicht. Der Roboter baut die Molekülmodelle zusammen, welche vorher in einem GUI ausgewählt worden sind. Mit 47 baubaren Molekülmodellen hat der Roboter meine Anfangserwartungen von nur acht baubaren Molekülmodellen weit übertroffen.

Um den Roboter in Aktion zu sehen, liegt dieser Arbeit eine DVD bei.

7. Zusammenfassung

Das Ziel dieser Maturitätsarbeit war, einen funktionstüchtigen Delta-Roboter zu konzipieren, zu konstruieren und zu programmieren, welcher verschiedene Molekülmodelle zusammensetzen kann. Der Benutzer sollte über ein GUI ein Molekülmodell auswählen können, welches dann vom Delta-Roboter automatisch gebaut wird.

Der Delta-Roboter ist auf einer Plattform an parallel konstruierten Aluminium-Trägern befestigt. Er ist aus LEGO-Technik Bauelementen gebaut und wird durch zwei LEGO Mindstorms NXTs gesteuert. Er besitzt Zahnradübersetzungen, um die Präzision der Motoren zu maximieren und ein ausgereiftes Greifsystem, um alle verschiedenförmigen Molekül-Teile greifen zu können.

Auf einer auf der Plattform befestigten LEGO-Platte befinden sich 14 Molekül-Teile auf Podesten, welche vom Roboter auf einer Vorrichtung in der Mitte zu 47 verschiedenen Molekülen zusammengebaut werden können.

Der Roboter wurde mit der graphischen Entwicklerplattform LabVIEW programmiert und interagiert mit einem GUI auf einem Computer, welcher den Roboter steuert. Sobald über das GUI ein Molekül durch ein Textfeld ausgewählt wurde, werden ein 2D- und 3D-Bild des Moleküls und wichtige Informationen über den Bauprozess auf dem GUI angezeigt.

Als das Programm auf die einzelnen Positionen der Molekül-Teile genau angepasst worden war, konnte der Roboter jedes der 47 Moleküle erfolgreich bauen. Auch wenn durch eine Zeitverzögerung beim Computer oder bei den NXTs ein Bauprozess mal nicht vollständig funktionierte, arbeitete der LEGO-Roboter sonst mit einer erstaunlichen Präzision.

Da das Anfangsziel erreicht wurde und mit 47 baubaren Molekülen meine Erwartungen mehr als übertroffen worden sind, betrachte ich dieses Projekt als erfolgreich.

8. Schlusswort

Abschliessend lässt sich sagen, dass für viele, ganz verschiedene Probleme einfachere und schwierigere Lösungen gesucht werden mussten, um diese Arbeit erfolgreich fertigstellen zu können.

Obwohl es Phasen des Zweifels gab, wenn etwas über längere Zeit nicht funktioniert hatte, ging meine Motivation nie verloren, und ich arbeitete stets weiter, bis ich schliesslich doch einen Lösungsweg fand. Neben dem Bauen und Programmieren eines Roboters habe ich auch gelernt habe, selber Lösungsansätze zu ganz verschiedenen Problemen zu entwickeln und umzusetzen.

Diese Maturitätsarbeit hat mir ausserordentlich Spass gemacht und meine Faszination für Technik und Maschinen nur noch intensiviert, so dass ich fest vorhave, im Herbst 2015 mit dem Studium der Maschineningenieurwissenschaften an der ETH Zürich zu beginnen.

9. Dank

An dieser Stelle würde ich gerne folgenden Personen danken:

Meiner betreuenden Lehrperson, Michael Liebich, für die tolle Unterstützung und dafür, dass er mich zu immer höheren Leistungen motiviert hat.

Meinen Eltern, welche mich immer unterstützt haben und meine schlechte Laune ertragen mussten, wenn etwas nicht funktioniert hat.

Dominik Mannhart (Klasse 6e), welcher mir seine Werkstatt zur Verfügung gestellt hat und mich immer mit LEGO-Teilen versorgt hat, falls mir welche gefehlt haben.

Stefan Kraft (Klasse 6e), welcher mir seinen NXT mit allen Motoren, Sensoren und Kabeln ausgeliehen hat.

10. Verwendete Anwendungen

Um diese Maturitätsarbeit zu realisieren, wurden die folgenden Programme verwendet:

-LabVIEW 2012 for Education von National Instruments

Programmierung des Roboters

-Pages '09 von Apple

Erstellen dieser Dokumentation

-Numbers '09 von Apple

Tabelle mit allen baubaren Molekülen (siehe Anhang, 14.2)

-iMovie '11 von Apple

Zusammenschneiden des Filmes über den Roboter

-Photoshop Elements von Adobe

Bildbearbeitung

-MarvinSketch 6.2.2 von ChemAxon

Benennung der Moleküle nach IUPAC und 2D-Bilder der Moleküle

-The Virtual Molecular Model Kit von CheMagic

3D-Bilder der Moleküle

-GitHub

Versionskontrolle der Dateien

11. Literatur- und Quellenverzeichnis, Nachschlagewerke

Tutorials

Grieder, Stefan, Dr.: Programmieren (für den Ti-89, Ti-Basic), Dossier, 2012

Programming for LEGO MINDSTORMS NXT - LabVIEW for LEGO MINDSTORMS, YouTube, 17. Mai 2011. Abgerufen im Mai, Juli und August 2014:
<https://www.youtube.com/watch?v=impPTmfm-Kg>

Programming Tutorial - LabVIEW for Education, YouTube, 17. Mai 2011. Abgerufen Mai, Juli und August 2014:

<https://www.youtube.com/watch?v=xpsQFvc7v5E>

LabView Basic Tutorial 1 (C-to-F Conversion), YouTube, 25. Februar 2011. Abgerufen im Mai, Juli und August 2014:

<https://www.youtube.com/watch?v=7xa6d5ckOMg>

LabView Basic Tutorial 2 (For Loop, Do While and Arrays), YouTube, 22. Februar 2011. Abgerufen im Mai, Juli und August 2014:

<https://www.youtube.com/watch?v=bt0x5J6r8Uo>

Tutorial: Arrays and Clusters, National Instruments, 2. Juli 2008. Abgerufen im Mai und August 2014:

<http://www.ni.com/white-paper/7571/en/>

Using Arrays in NI LabVIEW, YouTube, 16. November 2010. Abgerufen im August 2014:

https://www.youtube.com/watch?v=_GlQ1riWjPc

VI High 24: The Very Basics of How to Create an Array in LabVIEW, YouTube, 27. April 2012. Abgerufen im August 2014:

<https://www.youtube.com/watch?v=Qp1WpxFbqDs>

VI High 30: How to Manipulate Arrays by Indexing the Input of a LabVIEW For Loop, YouTube, 14. Juni 2012. Abgerufen im August 2014:

<https://www.youtube.com/watch?v=lHAIgWz91Rc>

LabVIEW Tutorial - Arrays, YouTube, 13. Dezember 2012. Abgerufen im August 2014:

<https://www.youtube.com/watch?v=LW1DoqQYJl8>

How to Clear a Chart or Graph Programmatically in LabVIEW, National Instruments, 22. Juli 2013. Abgerufen im September 2014:

<http://digital.ni.com/public.nsf/allkb/5B0B282E31E65E42862577A7005BC9AA?OpenDocument>

How To Open an Image File and Display it in LabVIEW, National Instruments, letzte abgerufene Version: 5. Dezember 2014. Abgerufen im November und Dezember 2014:

<http://digital.ni.com/public.nsf/allkb/02971A30F5D8FC6986256A0A004F11A0>

Boolesche Variable, Wikipedia, 30. November 2014. Abgerufen im Dezember 2014:

http://de.wikipedia.org/wiki/Boolesche_Variable

Molekülsystem

Molymod, offizielle Website von Molymod. Abgerufen im September 2014:
<http://www.molymod.com>

Molekülbaukasten Organische Chemie, Pearson Studium, September 2006. Abgerufen im September 2014:
<http://www.pearson.ch/1471/9783827372635/Molekuelbaukasten-Organische-Chemie.aspx>

Stäbchenmodell, Wikipedia, 27. Januar 2014. Abgerufen im September 2014:
<http://de.wikipedia.org/wiki/Stäbchenmodell>

Kalottenmodell, Wikipedia, 27. Januar 2014. Abgerufen im September 2014:
<http://de.wikipedia.org/wiki/Kalottenmodell>

Molekülbilder und Nomenklatur

Liebich, Michael: Schreibweise und Nomenklatur von Kohlenwasserstoffen, Dossier, 2013

MarvinSketch von ChemAxon für die 2D-Bilder des GUIs, Version 6.2.2. Abgerufen im November und Dezember 2014:
<http://www.chemaxon.com/products/marvin/marvinsketch/>

The Virtual Molecular Model Kit von CheMagic. Abgerufen im November und Dezember 2014:
<http://chemagic.com/JSmolVMK2.htm?rid=&echo=>

Roboter Modelle als Inspirationsquelle

CubeStormer II, YouTube, 13. Oktober 2011. Abgerufen April 2014:
https://www.youtube.com/watch?v=_d0LfkIut2M

Philips Medication Dispensing Service, Offizielle Website von Phillips. Abgerufen im Mai 2014:
<http://www.managemypills.com/content/>

NXT delta robot, YouTube, 20. August 2012. Abgerufen im Mai 2014:
<https://www.youtube.com/watch?v=z5KqdRmUOmg>

Mini Delta Robot with background noise to verify actual speed, YouTube, 27. Februar 2009. Abgerufen im Mai 2014:
<https://www.youtube.com/watch?v=Gv5B63HeF1E>

High Speed Pancake stacking with Flexpicker Robots, YouTube, 15. Juli 2009. Abgerufen im Mai 2014:
<https://www.youtube.com/watch?v=v9oeOYMRvuQ>

High Speed Adept Quattro Robot handling steel balls on conveyor, YouTube, 9. September 2008. Abgerufen im Juli 2014:
<https://www.youtube.com/watch?v=0-Kpv-ZOcKY>

Adept Quattro Parallel Robots, offizielle Website von Adept. Abgerufen im Juli 2014:

<http://www.adept.com/products/robots/parallel/quattro-s650h/general>

Roboter als Barkeeper - Robot bartender, YouTube, 5. September 2013. Abgerufen im Juni 2014:

<https://www.youtube.com/watch?v=i7anJjVJ2dY>

EWU Robotics - BlackJack Dealer, YouTube, 5. März 2014. Abgerufen im Juni 2014:

<https://www.youtube.com/watch?v=4YNbFXWoVN>

Informationen zu Robotern

Delta robot, Wikipedia, letzte abgerufene Version: 28. September 2014. Abgerufen im Mai 2014:

http://en.wikipedia.org/wiki/Delta_robot

The LEGO Group: Bedienungsanleitung Mindstorms NXT, 2006

Servomotor, Wikipedia, 16. September 2014. Abgerufen im Dezember 2014:

http://en.wikipedia.org/wiki/Delta_robot

Konversationen mit aussenstehenden Personen

-Vance Carter, Geschäftsführer EducaTec AG,
(Profil: <http://www.educatec.ch/Members/vcarter/>)

am 24-Stunden Wettbewerb 2014 von Helveticrobot in Chur, Januar 2014

-Sepios Team der ETH Zürich,
(Website: <http://www.sepios.org>)

an den ETH Studienwochen 2014, Juni 2014

-Simon D. Züst, MSc ETH ME, Research Assistant

an den Studieninformationstagen 2014 der ETH Zürich, September 2014

Maturitätsarbeit als Referenz

Hager, Pascal: Navigation eines LEGO™-Roboters mit Künstlicher Intelligenz, Gymnasium Thun Schadau, September 2007

Nachschatgewerke:

Orthographie:

Duden, offizielle Website von Duden:
<http://www.duden.de>

Wörterbuch:

dict.cc, offizielle Website der dict.cc GmbH
<http://www.dict.cc>

Bildquellen

Abb. 1: Robotikwettbewerb in Chur 2014

Abgerufen im Dezember 2014

<http://www.helveticrobot.ch/wettbewerbe/24h-wettbewerb/24h-wettbewerb-2014/>

Abb. 5: Bauplan eines gewöhnlichen Delta-Roboters

Abgerufen im Dezember 2014

<http://www.elmomc.com/capabilities/6.GMAS%20Multi%20Axis%20Motion/12.GMAS%20Delta%20Robot/GMAS-Delta-Robot.htm#page=page-1>

Abb. 6: Originalverpackung des LEGO Mindstorms NXTs

Abgerufen im Dezember 2014

<http://www.ebay.com/itm/like/350870584297?lpid=82>

Abb. 7: LEGO Mindstorms NXT mit allen Sensoren und Servomotoren

Abgerufen im Dezember 2014

<http://jaxenter.de/artikel/LEGO-MINDSTORMS-NXT-166552>

Abb. 8: Getriebe im Innern eines NXT Servomotors

Abgerufen im Dezember 2014

http://www.philohome.com/nxtmotor/motor1_3.jpg

Abb. 9: NXT Servomotor

Abgerufen im Dezember 2014

<http://shop.lego.com/en-US/Interactive-Servo-Motor-9842>

Abb. 11: Kubischer Elektromotor

Abgerufen im Dezember 2014

<http://www.philohome.com/motors/motorcomp.htm>

Abb. 18: Molymod-Molekülbaukasten

Abgerufen im Dezember 2014

<http://www.pearson.ch/bigcovers/9783827372635.aspx>

12. Rechtshinweis

Für alle erwähnten und rechtlich geschützten Marken- und Produktnamen gelten die rechtlichen Bestimmungen.

13. Unterzeichnete Erklärung

„Ich, Victor Fabrizio Klemm aus der Klasse 6e, erkläre hiermit, dass ich die vorliegende Arbeit „ΔELTAFOUR – Ein Delta-Roboter setzt Molekülmodelle zusammen“ selbstständig und ohne Benützung anderer als der angegebenen Quellen oder Hilfsmittel verfasst bzw. gestaltet habe.“

Ort, Datum: _____ Unterschrift: _____

14. Anhang

14.1 Ausdruck des gesamten Programms

Ein Zusatzblatt im A3-Format zeigt die endgültige Version des gesamten LabVIEW-Programms. Es ist zu beachten, dass alle Case Structures über ein Drop-down-Menu verfügen, welches beim Ausdruck nicht ersichtlich ist. Jede wählbare Möglichkeit enthält eine eigene Programmanweisung (zwei Case Structures haben sogar fast 50 Möglichkeiten im Drop-down-Menu), welche durch den Input der Case Structure bestimmt werden kann.

14.2 Tabelle der baubaren Moleküle

Auf der folgenden Tabelle sind alle wichtigen Informationen zu den 47 baubaren Molekülen aufgelistet.

Nr.	Name Deutsch	Name Englisch	erweiterte Summenformel	erw. SF verkürzt	Bewegungs-Sequenz
1	Ethan	ethane	CH3CH3		1,15,6,16,0
2	Propan	propane	CH3CH2CH3		10,17,1,15,12,16,0
3	Butan	butane	CH3CH2CH2CH3	CH3(CH2)2CH3	13,17,1,15,12,16,6,18,0
4	Pentan	pentane	CH3CH2CH2CH2CH3	CH3(CH2)3CH3	13,17,1,15,10,19,12,16,11,18,0
5	Propen	propene	CH3CHCH2		10,17,9,15,8,16,0
6	But-1-en	but-1-ene	CH3CH2CHCH2		13,17,9,15,8,16,6,18,0
7	Pent-1-en	pent-1-ene	CH3CH2CH2CHCH2	CH3(CH2)2CHCH2	13,17,9,15,10,19,8,16,12,18,0
8	Methylamin	methylamine	CH3NH2		5,15,6,16,0
9	Ethylamin	ethylamine	CH3CH2NH2		10,17,5,15,12,16,0
10	Propyl-1-amin	propyl-1-amine	CH3CH2CH2NH2	CH3(CH2)2NH2	13,17,5,15,12,16,6,18,0
11	Butyl-1-amin	butyl-1-amine	CH3CH2CH2CH2NH2	CH3(CH2)3NH2	13,17,5,15,10,19,12,16,11,18,0
12	Methanol	methanol	CH3OH		14,20,6,16,0
13	Ethanol	ethanol	CH3CH2OH		10,17,14,20,12,16,0
14	Propanol	propanol	CH3CH2CH2OH	CH3(CH2)2OH	13,17,14,20,12,16,6,18,0
15	Butanol	butanol	CH3CH2CH2CH2OH	CH3(CH2)3OH	13,17,14,15,10,19,12,16,11,18,0
16	Chlorehan	chloroethane	CH3CH2Cl		2,15,6,16,0
17	1-Chlorpropan	1-chloropropane	CH3CH2CH2Cl	CH3(CH2)2Cl	10,17,2,15,12,16,0
18	1-Chlorbutan	1-chlorobutane	CH3CH2CH2CH2Cl	CH3(CH2)3Cl	13,17,2,15,12,16,6,18,0
19	1-Chlorpentan	1-chloropentane	CH3CH2CH2CH2CH2Cl	CH3(CH2)4Cl	13,17,2,15,10,19,12,16,11,18,0
20	Iodethan	iodoethane	CH3CH2I		3,15,6,16,0
21	1-Iodpropan	1-iodopropane	CH3CH2CH2I	CH3(CH2)2I	10,17,3,15,12,16,0
22	1-Iodbutan	1-iodobutane	CH3CH2CH2CH2I	CH3(CH2)3I	13,17,3,15,12,16,6,18,0
23	1-Iodpentan	1-iodopentane	CH3CH2CH2CH2CH2I	CH3(CH2)4I	13,17,3,15,10,19,12,16,11,18,0
24	Bromethan	bromoethane	CH3CH2Br		4,15,6,16,0
25	1-Brompropan	1-bromopropane	CH3CH2CH2Br	CH3(CH2)2Br	10,17,4,15,12,16,0
26	1-Brombutan	1-bromobutane	CH3CH2CH2CH2Br	CH3(CH2)3Br	13,17,4,15,12,16,6,18,0
27	1-Brompentan	1-bromopentane	CH3CH2CH2CH2CH2Br	CH3(CH2)4Br	13,17,4,15,10,19,12,16,11,18,0
28	Ethenylamin	ethenylamine	CH2CHNH2		9,19,5,17,8,18,0
29	But-3-enyl-1-amin	but-3-enyl-1-amine	CH2CHCH2CH2NH2	CH2CH(CH2)2NH2	13,17,5,15,9,19,12,16,8,18,0
30	Ethenol	ethenol	CH2CHOH		14,17,9,19,8,21,0
31	But-3-en-1-ol	but-3-en-1-ol	CH2CHCH2CH2OH	CH2CH(CH2)2OH	13,17,14,20,9,19,12,22,8,21,0

32	3-Chlorprop-1-en	3-chloroprop-1-ene	CH2CHCH2Cl	2,17,9,19,8,18,0
33	5-Chlorpent-1-en	5-chloropent-1-ene	CH2CHCH2CH2CH2Cl	CH2CH(CH2)3Cl 13,17,2,15,9,19,12,16,8,18,0
34	3-Iodprop-1-en	3-iodoprop-1-ene	CH2CHCH2I	3,17,9,19,8,18,0
35	5-Iodopent-1-en	5-iodopent-1-ene	CH2CHCH2CH2CH2I	CH2CH(CH2)3I 13,17,3,15,9,19,12,16,8,18,0
36	3-Bromprop-1-en	3-bromoprop-1-ene	CH2CHCH2Br	4,17,9,19,8,18,0
37	5-Brompent-1-en	5-bromopent-1-ene	CH2CHCH2CH2CH2Br	CH2CH(CH2)3Br 13,17,4,15,9,19,12,16,8,18,0
38	Aminomethanol	aminomethanol	NH2CH2OH	5,17,7,19,12,18,0
39	3-Aminopropan-1-ol	3-aminopropan-1-ol	NH2CH2CH2CH2OH	NH2(CH2)3OH 13,17,5,15,7,19,12,16,11,18,0
40	Methandiol	methanediol	OHCH2OH	7,17,14,20,12,16,0
41	Propan-1,3-diol	propane-1,3-diol	OHCH2CH2CH2OH	OH(CH2)3OH 13,17,14,20,7,19,12,16,11,18
42	2-Chlorethan-1-ol	2-chloroethan-1-ol	CICH2CH2OH	2,17,7,19,12,18,0
43	4-Chlorbutan-1-ol	4-chlorobutan-1-ol	CICH2CH2CH2CH2OH	Cl(CH2)4OH 13,17,2,15,7,19,12,16,11,18,0
44	2-Iodethan-1-ol	2-idoethan-1-ol	ICH2CH2OH	3,17,7,19,12,18,0
45	4-Iodbutan-1-ol	4-iodobutan-1-ol	ICH2CH2CH2CH2OH	I(CH2)4OH 13,17,3,15,7,19,12,16,11,18,0
46	2-Bromethan-1-ol	2-bromoethan-1-ol	BrCH2CH2OH	4,17,7,19,12,18,0
47	4-Brombutan-1-ol	4-bromobutan-1-ol	BrCH2CH2CH2CH2OH	Br(CH2)4OH 13,17,4,15,7,19,12,16,11,18,0

14.3 Entwürfe und Zeichnungen

Auf den folgenden Seiten sind die wichtigsten Zeichnungen und Entwürfe abgedruckt.